



Pengembangan GAME AR (Augmented Reality)

Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech





Pengembangan GAME AR (Augmented Reality)

Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-623-8642-64-9 (PDF)



9

786238

642649

Pengembangan Game AR (Augmented Reality)

Penulis :

Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

ISBN : 978-623-8642-64-9

Editor :

Dr. Ir. Agus Wibowo, M.Kom, M.Si, M.M.

Penyunting :

Dr. Joseph Teguh Santoso, M.Kom.

Desain Sampul dan Tata Letak :

Irdha Yuniyanto, S.Ds., M.Kom.

Penebit :

Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Anggota IKAPI No: 279 / ALB / JTE / 2023

Redaksi :

Jl. Majapahit no 605 Semarang

Telp. (024) 6723456

Fax. 024-6710144

Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :

Universitas STEKOM

Jl. Majapahit no 605 Semarang

Telp. (024) 6723456

Fax. 024-6710144

Email : info@stekom.ac.id

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa ijin dari penulis

KATA PENGANTAR

Puji syukur kita panjatkan ke hadirat Tuhan Yang Maha Esa, atas rahmat dan karunia-Nya, sehingga buku yang berjudul *“Pengembangan Game AR (Augmented Reality)”* ini dapat diselesaikan dengan baik. Buku ini disusun sebagai upaya untuk memberikan wawasan dan pemahaman yang lebih mendalam mengenai konsep pengembangan game berbasis Augmented Reality (AR) di platform iOS, yang semakin berkembang pesat dalam dunia teknologi dan pengembangan aplikasi mobile.

Pengembangan game AR merupakan salah satu inovasi yang paling menarik dalam industri game dan teknologi mobile, memungkinkan para pengembang untuk menciptakan pengalaman bermain yang lebih imersif dan interaktif dengan menggabungkan dunia nyata dan digital. Dengan menggunakan platform iOS dan framework ARKit, pengembang dapat memanfaatkan potensi teknologi AR untuk menciptakan game yang tidak hanya menghibur, tetapi juga memberikan pengalaman unik bagi para pemain.

Buku ini hadir sebagai panduan praktis bagi para pengembang yang ingin memulai perjalanan mereka dalam mengembangkan game berbasis AR. Diharapkan, dengan pemahaman yang diperoleh dari buku ini, pembaca dapat menguasai dasar-dasar pengembangan game AR, serta memanfaatkan teknologi ini untuk menciptakan aplikasi yang inovatif dan menarik.

Buku ini terdiri dari delapan bab yang dirancang secara sistematis untuk membimbing pembaca dari tahap awal hingga tahap akhir dalam pengembangan game AR: Bab 1 membahas dasar-dasar memulai pengembangan game AR dengan Unity, termasuk persyaratan sistem dan pengaturan awal. Bab 2 memperkenalkan konsep dasar Augmented Reality dan bagaimana memulai proyek AR. Bab 3 fokus pada penggunaan Unity ARKit, memberikan wawasan tentang fitur-fitur penting yang dapat dimanfaatkan. Bab 4 menjelaskan proses pengujian tabrakan dan pencahayaan, aspek krusial dalam menciptakan pengalaman bermain yang realistis. Bab 5 memberikan panduan praktis dalam membuat game AR dari awal hingga akhir. Bab 6 memperkenalkan kontrol sentuh untuk meningkatkan interaktivitas dalam permainan. Bab 7 membahas deteksi bidang dan awan titik, dua elemen penting dalam pengembangan AR. Bab 8 menyajikan langkah-langkah akhir dalam menyelesaikan proyek game AR.

Setiap bab dilengkapi dengan penjelasan yang jelas dan contoh praktis yang diharapkan dapat memudahkan pembaca dalam memahami konsep-konsep yang disampaikan. Dengan pendekatan yang terstruktur ini, kami berharap pembaca dapat mengembangkan keterampilan mereka dan menciptakan game AR yang menarik dan inovatif.

Semoga buku ini dapat memberikan manfaat, inspirasi, dan menjadi langkah awal bagi pembaca untuk mengeksplorasi dunia pengembangan game AR di platform iOS. Terima Kasih.

Semarang, Desember 2024

Penulis

Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

DAFTAR ISI

Halaman Judul	i
Kata Pengantar	ii
Daftar Isi	iii
BAB 1 MEMULAI PENGEMBANGAN GAME AR DENGAN UNITY	1
1.1 Unity3D	1
1.2 Persyaratan Unity	2
1.3 Menyiapkan Unity	6
1.4 Persyaratan Pengembangan Ios	8
BAB 2 MEMULAI DENGAN AR	10
2.1 Panduan Untuk Membangun Game AR Di Platform Ios	10
2.2 Memasang Unity Arkit.....	13
2.3 Tata Letak Editor Unity	14
2.4 Ubah Ukuran Area	18
2.5 Skala Ikon	25
2.6 Gameobject Induk Dan Anak	28
2.7 Opsi Scene View	32
2.8 Tampilan Game	33
2.9 Manual Unity	37
BAB 3 UNITY ARKIT	39
3.1 Membuat Adegan	39
3.2 Kamera.....	41
3.3 Transformasi	42
3.4 Pengujian	47
3.5 Melacak Pergerakan Ponsel	49
3.6 Memperkenalkan Visual Inertial Odometry	55
BAB 4 PENGUJIAN TABRAKAN DAN PENCAHAYAAN	64
4.1 Pengujian Tabrakan	64
4.2 Skala.....	66
4.3 Transformasi	67
4.4 Skrip Uji Tabrakan Editor	69
4.5 Menambahkan Komponen	70
4.6 Pencahayaan	74
BAB 5 MEMBUAR GAME AR	80
5.1 Fugu Bowlgame AR	80
5.2 Membuat Aset AR	82
5.3 Membuat Bidang (Sementara)	89
5.4 Mengimpor File OBJ Ke Unity	92
5.5 Mengedit Skrip Playercontroller	100
BAB 6 MEMPERKENALKAN SENTUHAN	106
6.1 Menguji Game Dalam AR	106
6.2 Beberapa Tantangan	107

6.3	Menerapkan Kontrol Sentuh	109
6.4	Pengujian	114
BAB 7	MENAMBAHKAN DETEKSI BIDANG DAN AWAN TITIK	117
7.1	Membuat Gameobject Bidang Yang Dihasilkan	117
7.2	Menyiapkan Kamera Utama	119
7.3	Menambahkan Unity Remote Connection	123
7.4	Pengujian Pada Perangkat Ios	127
7.5	Pencahayaan	129
BAB 8	LANGKAH AKHIR	132
8.1	Membuat Prefab	132
8.2	Menambahkan Lebih Banyak Pin Bowling	134
8.3	Membuat Instansi Gameobject Saat Runtime	135
8.4	Membuat Game Manager	138
8.5	Pengujian	140
Daftar Pustaka	142

BAB 1

MEMULAI PENGEMBANGAN GAME AR DENGAN UNITY

Dalam buku ini, kita akan mempelajari cara membuat Game Augmented Reality (atau AR) menggunakan perangkat lunak pengembangan game dari Unity (Unity3D 2018 atau, lebih umum, Unity). Dalam bab ini, kita akan membahas proses mengunduh dan menginstal Unity serta mempelajari beberapa alat yang disediakan Unity. Kita juga akan menginstal proyek AR yang sudah ada dari Unity Asset Store dan menjelajahi beberapa fitur game ini. Dalam Bab 2, kita akan menginstal ARKit dan memberikan gambaran umum tentang antarmuka pengguna Unity. Dalam Bab 3, kita akan mulai menggunakan Unity ARKit dan menggunakan beberapa fungsi utama.

Saya juga akan memberikan gambaran umum dasar tentang odometri inersia visual dan apa artinya ini untuk membuat proyek AR. Kita akan membuat adegan dasar untuk menggunakan dan menguji Unity ARKit. Dalam Bab 4, kita akan menggunakan beberapa fungsi yang lebih canggih di Unity ARKit seperti pengujian hit dan pencahayaan adegan AR. Terakhir, di Bab 5, kita akan menggabungkan semuanya dan membuat game AR menggunakan Unity ARKit. Buku ini ditulis untuk pemula yang belum memiliki pengalaman sebelumnya menggunakan Unity atau membuat game.

Bab-babnya telah disiapkan secara berurutan untuk membantu mempelajari setiap langkah. Namun, jika Anda membaca buku ini dan sudah tahu tentang Unity atau membuat game, silakan lewati bab mana pun yang menurut Anda sudah Anda ketahui.

1.1 UNITY3D

Mesin gim Unity adalah alat pengembangan gim lintas platform untuk membuat gim 2D dan 3D. Istilah lintas platform dapat memiliki arti yang berbeda bagi orang yang berbeda. Karena Unity dapat digunakan di macOS, Windows, atau Linux, ini dapat dianggap sebagai alat pengembangan lintas platform. Namun, karena Unity dapat digunakan untuk mengembangkan gim untuk konsol gim, komputer pribadi, peramban web, perangkat seluler, sistem VR, dll., ini juga dapat menjadi alasan mengapa Unity dianggap sebagai alat pengembangan lintas platform. Unity dapat digunakan untuk membuat gim 3D, yaitu gim tersebut tampak beroperasi dalam ruang 3D (memiliki X, Y, dan Z).

Unity juga dapat digunakan untuk membuat gim 2D. Baru-baru ini, Unity telah digunakan untuk membuat gim atau simulasi VR dan AR. Dalam buku ini, kami akan menggunakan versi terbaru Unity, yang saat ini adalah 2018.1. Namun, seperti kebanyakan perangkat lunak (dan banyak perangkat keras), Unity terus-menerus memperkenalkan fitur dan fungsi baru, dan saat buku ini dicetak, mungkin ada versi Unity yang lebih baru yang tersedia. Ketika Unity membuat perubahan kecil pada perangkat lunak mereka, mereka biasanya akan menambahkan nomor (seperti 2018.1.1).

Ketika pembaruannya sedikit lebih substansial, maka nomor versinya akan berubah (seperti 2018.2). Ketika Unity biasanya membuat perubahan besar, maka nomor versinya akan

berubah sepenuhnya (Unity 1, 2, 3, 4, 5). Pada bulan Juli 2017, Unity mengubah sistem penomoran versi ke tahun rilis (2017 dan sekarang 2018).

1.2 PERSYARATAN UNITY

Sebelum Anda mulai belajar membuat game, Anda perlu mengunduh Unity dan menginstalnya di Mac Anda. Meskipun memungkinkan untuk membuat game untuk perangkat iOS dengan Unity yang terinstal di Komputer Pribadi Windows, Anda perlu menggunakan perangkat lunak bernama Xcode untuk mem-port kode Unity agar dapat berjalan di Mac atau perangkat iOS. Saat ini, Xcode hanya tersedia di Mac.

Jadi, jika Anda memiliki PC Windows, maka pada tahap tertentu Anda perlu menggunakan Mac untuk mem-port game tersebut. Di seluruh buku ini, saya akan menggunakan Mac; jika Anda menggunakan PC Windows, maka banyak petunjuk atau arahan mungkin tidak berlaku untuk Anda.

Mempersiapkan Mac Anda

Untuk pengembangan iOS menggunakan Unity, Anda memerlukan Mac yang menjalankan Lion atau Mountain Lion dari Mac OS X 10.9 atau yang lebih tinggi dan Xcode 7.0 atau yang lebih tinggi. Unity 2018 mungkin masih berjalan di beberapa sistem lama, tetapi Anda memerlukan Xcode versi terbaru, yang, seperti disebutkan, diperlukan untuk pengembangan iOS. Versi terbaru Xcode biasanya mendukung versi iOS yang lebih baru. Pada saat tulisan ini dibuat, versi Xcode saat ini adalah versi 9, yang akan saya gunakan di seluruh buku ini.

Mendaftar

Saya sangat menyarankan untuk mengunjungi situs web Apple Developer (<https://developer.apple.com/>) dan mendaftar sebagai pengembang iOS. Meskipun ini bukan persyaratan mutlak dari buku ini, jika Anda ingin menerbitkan game di App Store, maka Anda harus menjadi Pengembang Apple yang terdaftar. Proses pendaftaran sebagai Pengembang Apple dapat memakan waktu cukup lama, terutama jika Anda mendaftarkan perusahaan. Langkah pertama adalah mendaftar sebagai Pengembang Apple (yang saat ini gratis), kemudian setelah Anda terdaftar, langkah berikutnya adalah mendaftar sebagai pengembang iOS.

Unduh Xcode

Anda tidak akan memerlukan Xcode hingga nanti di buku ini, tetapi akan sangat bermanfaat untuk mengunduh dan menginstal Xcode. Anda dapat menemukan versi terbaru Xcode di situs web Pengembang Apple (<https://developer.apple.com/>).

Unduh Unity

Sekarang adalah waktu yang tepat untuk menginstal Unity. Buka situs web Unity di <https://unity3d.com> lalu pilih Dapatkan Unity atau ketik <https://store.unity.com/>. Di halaman ini, Anda akan menemukan rilis Unity terbaru (pada saat penulisan ini, 2018.1). Anda juga dapat menemukan rilis Unity sebelumnya di situs web Unity. Meskipun hanya ada satu aplikasi Unity, Anda dapat berlangganan berbagai opsi lisensi, tergantung pada kebutuhan dan ukuran perusahaan Anda (jika Anda memilikinya). Tiga opsi lisensi saat ini adalah Personal, Plus, dan

Pro.

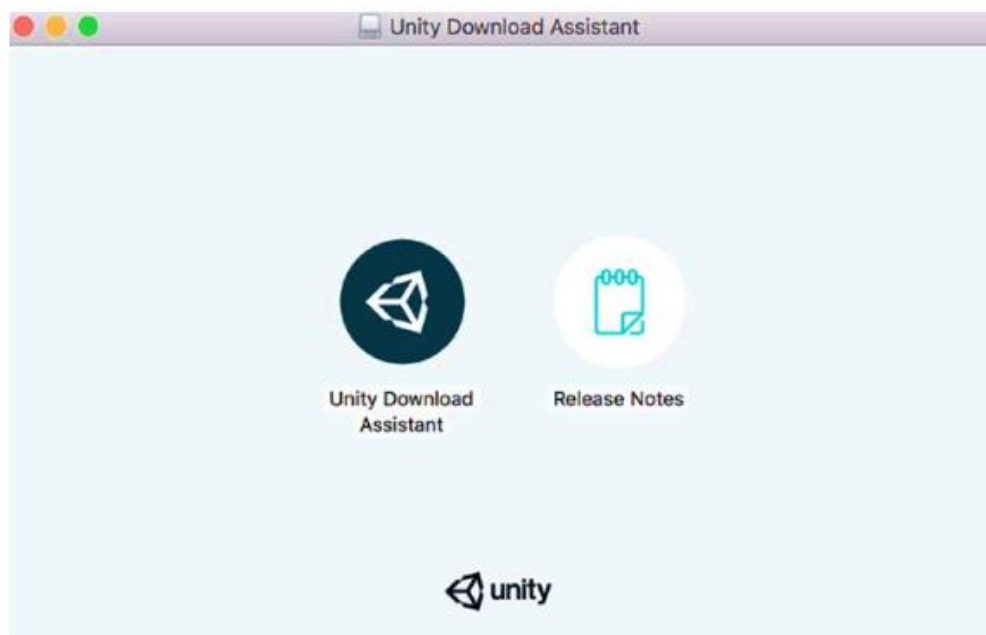
Untuk memulai proses pengunduhan, klik tombol opsi langganan yang sesuai dengan kebutuhan Anda (pada saat penulisan ini, ini akan menjadi Coba Personal, Dapatkan Plus, atau Jadi Pro). File tersebut berukuran sekitar 1 GB, jadi mungkin perlu waktu cukup lama untuk mengunduhnya. Sementara Anda menunggu, dan Anda berada di situs web Unity, luangkan waktu untuk memeriksa beberapa game dan demo yang telah dipublikasikan, situs komunitas, dan forum pengguna. Ini akan sangat berguna selama pengembangan game menggunakan Unity.

Instal Unity

File yang Anda unduh dari Unity adalah Penginstal Unduhan, yang pada saat penulisan ini bernama UnityDownloadAssistant.

Menjalankan Download Assistant

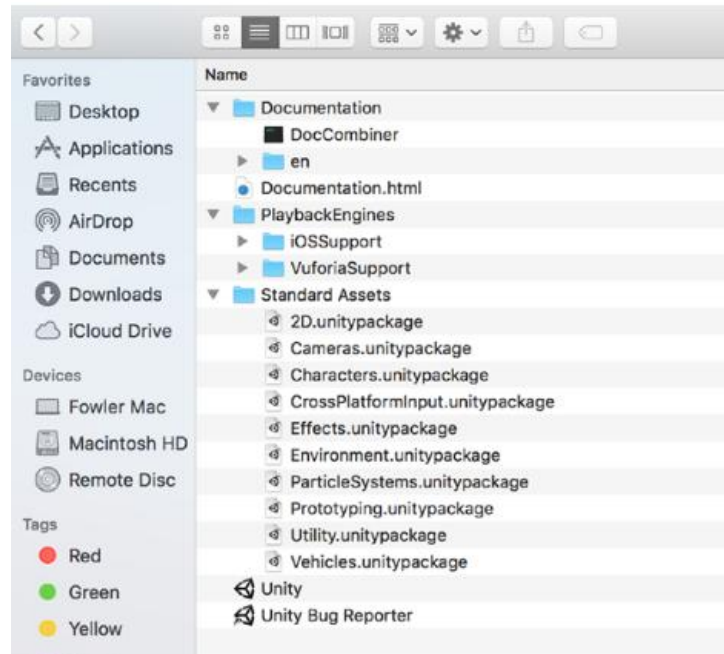
Setelah file UnityDownloadAssistant diunduh, klik dua kali file tersebut untuk menjalankan Unity Download Assistant. Klik dua kali ikon Unity Download Assistant untuk mulai menginstal Unity (Gambar 1.1).



Gambar 1.1 Unity Download Assistant

Penginstal akan memproses siklus penginstalan, dan setelah selesai, folder Unity akan ditempatkan di folder Aplikasi (kecuali Anda memilih lokasi yang berbeda). Jika Anda telah menginstal versi Unity sebelumnya, penginstalan versi baru dapat menggantikan versi sebelumnya. Sebaiknya ganti nama folder versi sebelumnya sebelum menginstal versi baru (misalnya, Unity2017). Dengan cara ini, Anda tetap dapat menggunakan kedua versi Unity.

Folder penginstalan Unity berisi aplikasi Unity dan beberapa folder dan aplikasi pendukung (Gambar 1.2).



Gambar 1.2 Folder Instalasi Unity

Salah satu file terpenting dalam folder Unity adalah aplikasi Unity, yang akan menyediakan alat yang digunakan untuk membuat dan menguji game Anda. Aplikasi ini terkadang disebut sebagai Unity Editor, yang berbeda dari aplikasi lain, Unity Runtime Engine (juga dikenal sebagai Unity Player). Unity Runtime Engine terintegrasi ke dalam versi final, yang akan memungkinkan game dimainkan pada perangkat keras target. Ketika saya merujuk ke Unity, saya biasanya merujuk ke Unity Editor. Terkadang saya akan merujuk ke perusahaan Unity Technologies sebagai Unity.

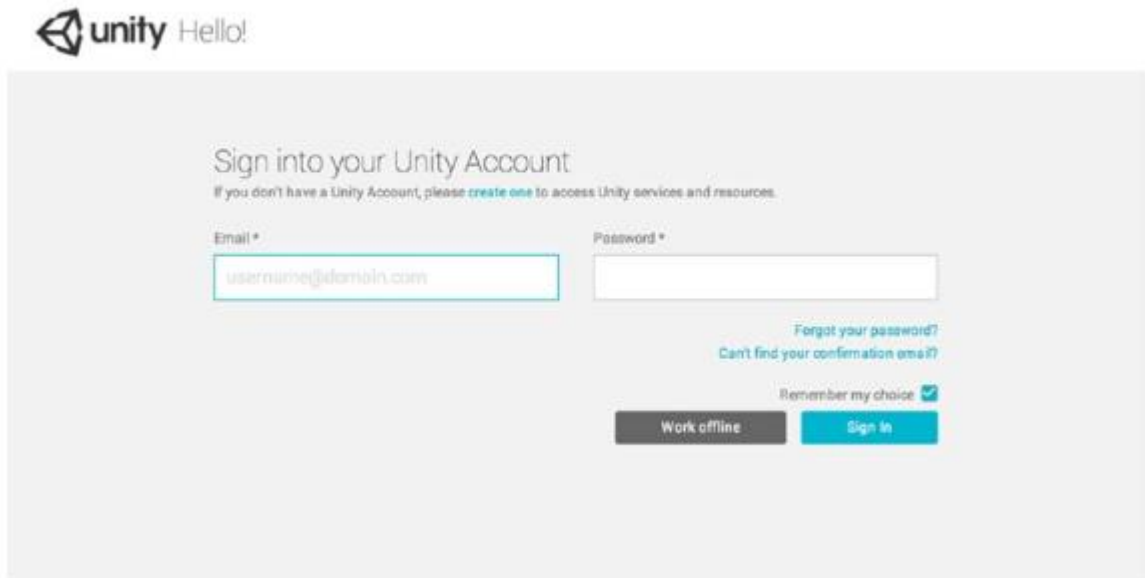
Namun, mudah-mudahan konteksnya jelas. Folder Dokumentasi berisi Panduan Pengguna, Referensi Komponen, dan dokumen Referensi Skrip. Dokumen-dokumen ini juga tersedia di situs web Unity (pilih tautan Pelajari). Semua file ini adalah dokumen HTML dan dapat dibuka di peramban web dari sistem menu Bantuan Unity, atau dapat dibuka langsung dengan mengklik dua kali file tersebut. Folder Aset Standar berisi beberapa file dengan ekstensi file .unityPackage. Ini adalah berkas paket Unity yang berisi kumpulan aset Unity, yang dapat diimpor ke Unity. Anda juga dapat membuat Aset Standar Anda sendiri dan mengeksport aset ini ke berkas paket. Ada juga Aplikasi Pelapor Bug Unity.

Aplikasi ini biasanya dijalankan langsung dari dalam Editor Unity menggunakan fungsi Laporkan Bug. Namun, aplikasi ini dijalankan langsung dari folder instalasi Unity. Jika Anda mengunduh Proyek Contoh dengan instalasi Unity, pastikan untuk membukanya di Unity. Jika Anda tidak mengunduhnya saat instalasi, Anda masih dapat mengunduhnya kapan saja.

Selamat Datang

Setelah Unity selesai diinstal (dan bersiaplah untuk menunggu beberapa saat), layar selamat datang editor Unity akan muncul dengan Jendela Unity Hello! (Gambar 1.3). Jendela Unity Hello! adalah tempat Anda masuk ke akun Unity (jika Anda memilikinya). Jika Anda tidak memiliki akun Unity, pilih tautan buat akun. Jika saat ini Anda tidak terhubung ke Internet,

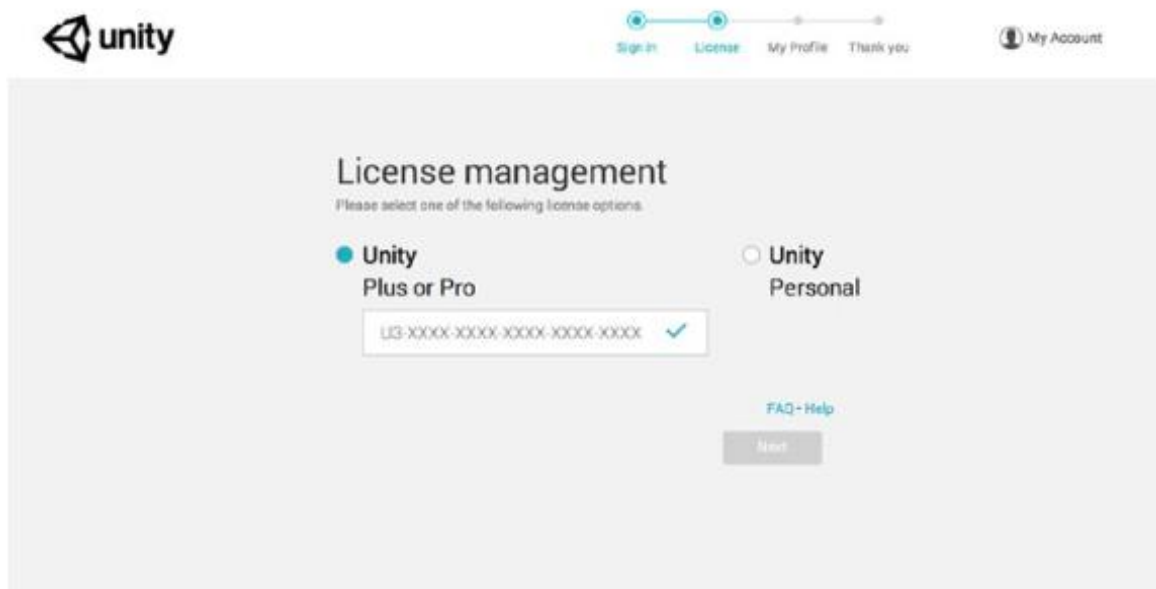
Anda dapat bekerja secara offline dengan memilih tombol Bekerja offline.



Gambar 1.3 Layar Unity Hello!

Jendela Unity Hello! akan muncul saat Anda memulai Unity

Saya sangat menyarankan untuk membuat akun Unity, jika Anda belum membuatnya. Setelah masuk untuk pertama kalinya, Anda akan melihat layar Manajemen lisensi. Jika Anda telah membayar untuk versi Unity berlisensi, masukkan nomor seri lisensi Anda di kotak dialog. Jika Anda ingin menggunakan versi Unity gratis, pilih tombol radio Unity Personal (Gambar 1.4).



Gambar 1.4 Layar Manajemen Lisensi Unity

1.3 MENYIAPKAN UNITY

Sebelum kita mulai membuat game dengan Unity, ini saat yang tepat untuk meninjau beberapa opsi dan fitur administratif Unity.

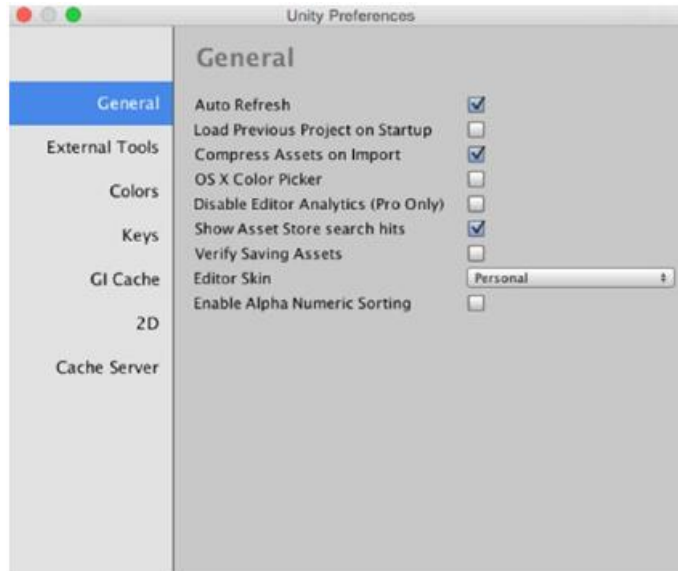
Mengubah Tampilan (Versi Pro)

Jika Anda telah membeli lisensi Pro Unity, Anda akan dapat memilih antara tampilan terang atau gelap. Jika Anda menggunakan Unity versi gratis, Anda hanya akan melihat Tampilan Terang.



Gambar 1.5 Menu Preferensi Unity

Karena sebagian besar pengembang game pemula menggunakan Unity versi gratis, saya akan menggunakan tampilan terang untuk tangkapan layar. Tampilan terang juga menghasilkan tangkapan layar yang lebih baik untuk versi buku saku ini. Jika Anda memiliki versi Pro dan ingin mengubah tampilan, pilih Preferensi di menu Unity (Gambar 1.5). Dengan membuka menu Preferensi, Anda dapat mengubah tampilan dari Gelap ke Terang atau Terang ke Gelap (Gambar 1-6). Jika Anda menggunakan Unity edisi Personal, Anda akan terjebak dengan tampilan Terang.

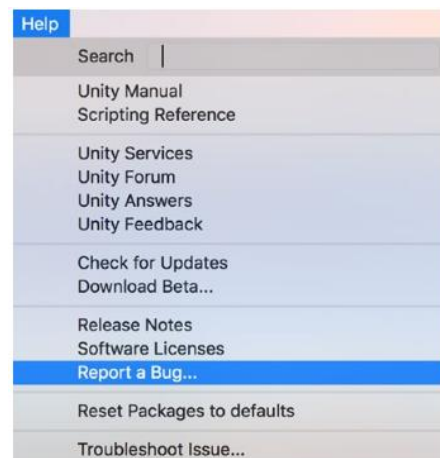


Gambar 1.6 Menu Preferensi Umum Di Editor Unity

Saat Anda membuka menu Preferensi Umum, saya sarankan untuk memastikan opsi Muat Proyek Sebelumnya saat Memulai tidak dipilih. Ini akan memastikan bahwa Unity memuat dialog pemilihan proyek saat memulai. Ini akan memastikan Anda terhindar dari memperbarui versi proyek yang salah atau memperbarui versi Unity yang Anda gunakan sebelum Anda siap.

Melaporkan Masalah

Jika Anda terus menggunakan Unity selama beberapa tahun, Anda akan menemukan beberapa bug (baik yang nyata maupun yang dibayangkan). Saya telah menggunakan Unity sejak versi 1.6 dan telah menemukan beberapa bug dengan Unity. Bug perangkat lunak tidak hanya terjadi pada Unity. Mesin pengembangan game adalah perangkat lunak yang kompleks dan Unity tentu menghargai dan menghargai laporan bug. Jika bug tidak dilaporkan, maka sulit bagi Unity untuk memperbaikinya. Aplikasi Pelapor Bug Unity menyediakan fitur ini. Seperti yang disebutkan sebelumnya, Report a Bug Reporter tersedia di folder instalasi Unity dan tersedia dari menu Help di Unity Editor (Gambar 1.7).



Gambar 1.7 Opsi Report A Bug Di Menu Help

Gambar 1.8 Jendela Unity Bug Reporter

Memilih opsi Report, a Bug pada menu bantuan atau mengklik dua kali aplikasi Report a Bug di folder instalasi Unity akan membuka aplikasi Unity Report a Bug (Gambar 1.8). Aplikasi ini menyediakan opsi menu dan kotak dialog bagi pengguna untuk menentukan masalah yang terkait, seberapa sering masalah terjadi, judul bug, detail bug, dan opsi untuk melampirkan file relevan yang akan membantu memperbaiki bug. Unity Bug Reporter mengharuskan pengguna untuk menentukan alamat email sehingga tim di Unity dapat menanggapi laporan bug.

1.4 PERSYARATAN PENGEMBANGAN IOS

Sebelumnya di bab ini, saya menyarankan agar mengunduh Xcode dan mendaftar di Apple Developer Program. Jika Anda belum mengunduhnya, sekarang adalah saat yang tepat untuk berhenti dan melakukannya. Persyaratan perangkat keras dan perangkat lunak untuk pengembangan iOS dan detail tentang Apple Developer Program tercantum di halaman dukungan Pengembang Apple (<https://developer.apple.com/>). Anda juga dapat menemukan persyaratan dan halaman unduhan untuk Xcode di <https://developer.apple.com/Xcode/>.

Situs Web Unity

Seiring dengan meningkatnya fitur dan fungsi Unity, Unity Technologies juga meningkatkan keluasan dan kedalaman konten. Ada banyak informasi di situs web ini, tetapi saya sarankan untuk melihat bagian Tanya Jawab Umum (<https://unity3d.com/unity/faq>). Ada

juga beberapa tutorial, dokumentasi, dan video hebat yang akan membantu Anda mempelajari cara membuat game di Unity (<https://unity3d.com/unity/faq>).

Komunitas Unity

Di menu Bantuan, terdapat tautan ke situs komunitas Unity resmi. Ini termasuk Forum Unity resmi (<https://forum.unity.com>), yang dimoderatori oleh staf Unity. Ini adalah sumber yang bagus untuk semua pengembang game. Di menu ini juga terdapat tautan ke Unity Answers, yang menggunakan format Stack Exchange dan menyertakan beberapa kontrol (atau moderasi) atas pertanyaan dan jawaban. Situs umpan balik Unity (<https://feedback.unity3d.com>) memungkinkan pengembang untuk meminta dan memberikan suara pada kemungkinan fitur di masa mendatang.

Selamat datang di Unity.

BAB 2

MEMULAI DENGAN AR

2.1 PANDUAN UNTUK MEMBANGUN GAME AR DI PLATFORM IOS

Setelah pembaca sudah memiliki gambaran yang baik tentang apa itu *Augmented Reality* (atau AR). Karena ada beberapa kebingungan tentang perbedaan antara istilah *Virtual Reality* (atau VR), *Augmented Reality*, dan *Mixed Reality*, saya pikir mungkin ada baiknya mencoba menjelaskan apa yang saya maksud dengan AR dalam buku ini. *Virtual Reality* (VR) adalah lingkungan buatan komputer yang mensimulasikan pengalaman melalui indra dan persepsi. Tidak seperti sistem komputer tradisional, sistem VR menempatkan pengguna di dalam sebuah pengalaman. Alih-alih melihat layar di depan mereka, pengguna tenggelam dan dapat berinteraksi dengan dunia 3D. Sekarang, mari kita lihat *Augmented Reality* (AR).

Konsensus umum adalah bahwa AR didefinisikan sebagai tampilan langsung dari lingkungan dunia nyata yang elemen-elemennya "ditambah" oleh informasi yang dihasilkan komputer. Perbedaan utama antara VR dan AR adalah bahwa AR mencakup tampilan langsung dari lingkungan nyata. Sistem VR biasanya tidak mencakup tampilan langsung dari lingkungan nyata. Headset VR tertutup sepenuhnya, dan tampilannya sepenuhnya dihasilkan komputer. *Mixed Reality* (MR, meskipun akronimnya jarang digunakan), adalah istilah yang paling banyak digunakan oleh Microsoft untuk membedakan HoloLens mereka. Saya merasa bahwa *Mixed Reality* adalah bentuk lain dari AR.

Namun, masih ada beberapa perdebatan tentang hal ini. Ketika saya bertanya kepada teman baik saya di Microsoft, mereka merasa bahwa *Mixed Reality* berada di antara AR dan VR dan sepenuhnya mengintegrasikan objek digital ke dalam dunia Anda, membuatnya tampak seolah-olah objek tersebut benar-benar ada. Mungkin Anda terkejut bahwa istilah VR dan AR telah ada selama bertahun-tahun. Meskipun ada perdebatan yang cukup panjang tentang penggunaan pertama istilah VR (sebagian besar karena definisi istilah yang disepakati), ada beberapa konsensus umum bahwa istilah tersebut pertama kali digunakan sekitar tahun 1950-an saat penulis merujuk pada sistem atau lingkungan yang sepenuhnya imersif.

Di seluruh Militer AS dan NASA, produsen pesawat menggunakan sistem AR untuk pelatihan, penelitian, dan pengembangan. Namun, baru pada tahun 2016 kita melihat sistem AR konsumen pertama yang tersedia secara komersial, Oculus Rift. Oculus Rift diproduksi oleh Oculus VR (yang akhirnya dibeli oleh Facebook seharga \$2.000.000.000). Referensi ke AR juga telah dibuat selama bertahun-tahun. Kesulitan dalam menyetujui kapan referensi pertama ke AR dibuat juga karena definisi yang disepakati tentang apa itu AR. Pada tahun 1989, George Douglas menulis tentang sistem kendali dan panduan teleskop astronomi yang digerakkan komputer dengan tampilan grafik koordinat langit dan bidang bintang yang tumpang tindih yang tampaknya merupakan sistem AR pertama.

Baru-baru ini, ada beberapa perkembangan menarik dalam AR. Meskipun ada beberapa sistem yang bersifat hak milik, perkembangan pertama yang paling menonjol adalah diperkenalkannya Pokémon Go untuk ponsel Android dan iOS. Dengan menggunakan

kemampuan geolokasi dan kamera terintegrasi pada ponsel, pengguna dapat melihat objek virtual muncul di layar seolah-olah berada di dunia nyata. Dalam buku ini, saya akan fokus mengembangkan permainan AR untuk iOS. Saya memilih iOS karena sejumlah alasan. Pertama, ada banyak perangkat yang menggunakan iOS. Yang lebih penting, Apple berinvestasi dalam pengembangan AR baik dalam perangkat keras maupun perangkat lunak.

Dengan iOS 11, Apple menyertakan ARKit. ARKit memudahkan kita membuat permainan dan simulasi AR yang menempatkan objek virtual ke dalam lingkungan pengguna. Dengan menggabungkan informasi dari sensor gerak perangkat dengan data dari kameranya, ARKit dapat membantu iPhone atau iPad menganalisis lingkungan sekitar. Apple juga meningkatkan kemampuan kamera pada iPhone 8 dan iPhone X. Kamera pada iPhone 8 dan iPhone X telah dirancang untuk mengambil gambar dalam cahaya redup dan video 60-fps. Stabilisasi gambar optik ganda pada iPhone X dan stabilisasi gambar optik yang ditingkatkan pada iPhone 8 juga memberikan kejernihan visual yang lebih baik.

Fitur perangkat keras dan perangkat lunak ini membantu membuat permainan tampak lebih alami di lingkungan pengguna. Apple juga berinvestasi besar dalam meningkatkan kemampuan AR pada iPhone dan iPad masa depan mereka. Unity baru-baru ini memperkenalkan Unity ARKit di Unity Store. Hal ini memudahkan kita membuat game AR untuk iOS. Oleh karena itu, di seluruh buku ini, saya akan menggunakan Unity ARKit. Jadi sekarang adalah saat yang tepat untuk menginstal versi terbaru Unity ARKit. Persyaratan minimum untuk Unity ARKit adalah sebagai berikut:

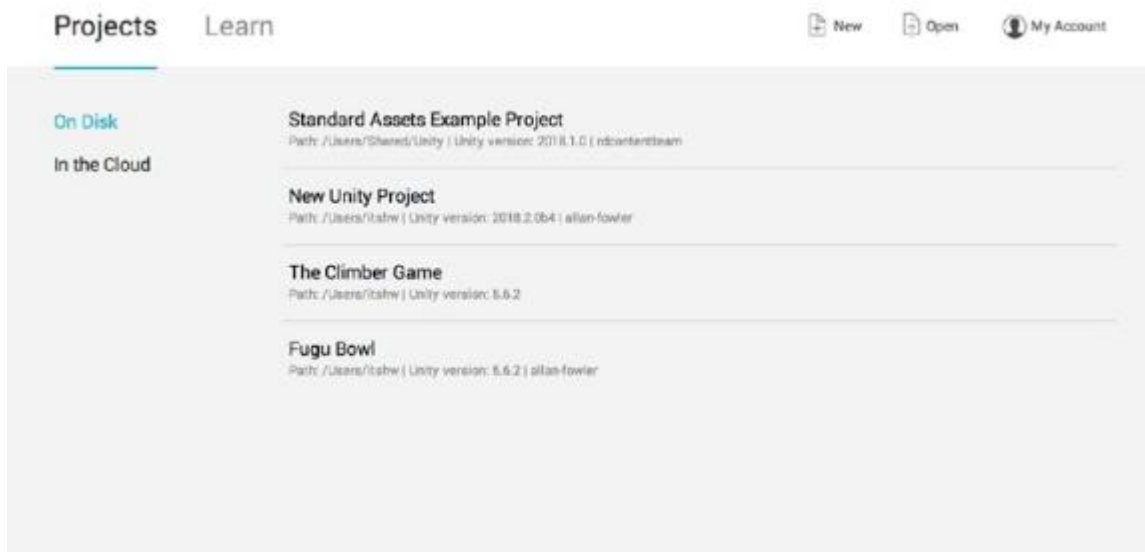
Perangkat iOS yang mendukung ARKit dan memiliki versi terbaru iOS 11.3 atau yang lebih tinggi.

Mac dengan macOS 10.13 (High Sierra) atau yang lebih tinggi.

Unity versi 2017.1 atau yang lebih tinggi.

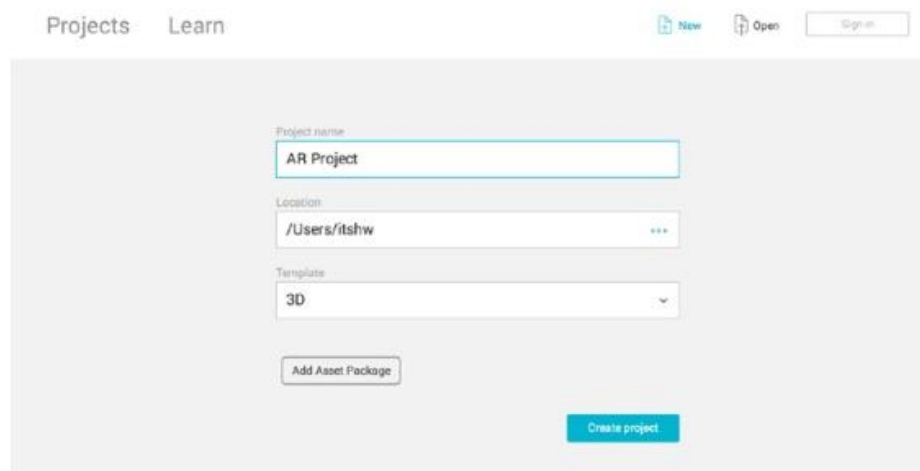
Versi terbaru XCode 9.3 (atau yang lebih tinggi) dari situs web Pengembang Apple (memerlukan macOS 10.13).

Sekarang setelah Anda menginstal Unity, saatnya untuk mengenal berbagai elemen Unity ARKit yang akan kita gunakan untuk membuat dan menguji game kita. Sementara kita menjelajahi berbagai komponen Unity ARKit, saya akan membahas beberapa prinsip teknis dalam mengembangkan game AR. Karena ini adalah buku pengantar, saya tidak akan membahas terlalu dalam detail teknisnya dan mencoba untuk tetap membahasnya secara umum. Editor saya meminta rekomendasi buku AR tambahan, jadi saya mungkin akan tetap membahas konten yang lebih mendalam atau teknis untuk buku kedua tentang AR. Sekarang mari kita mulai Unity dan buat proyek baru. Saat Anda memulai Unity, Anda akan melihat layar Proyek (Gambar 2.1).



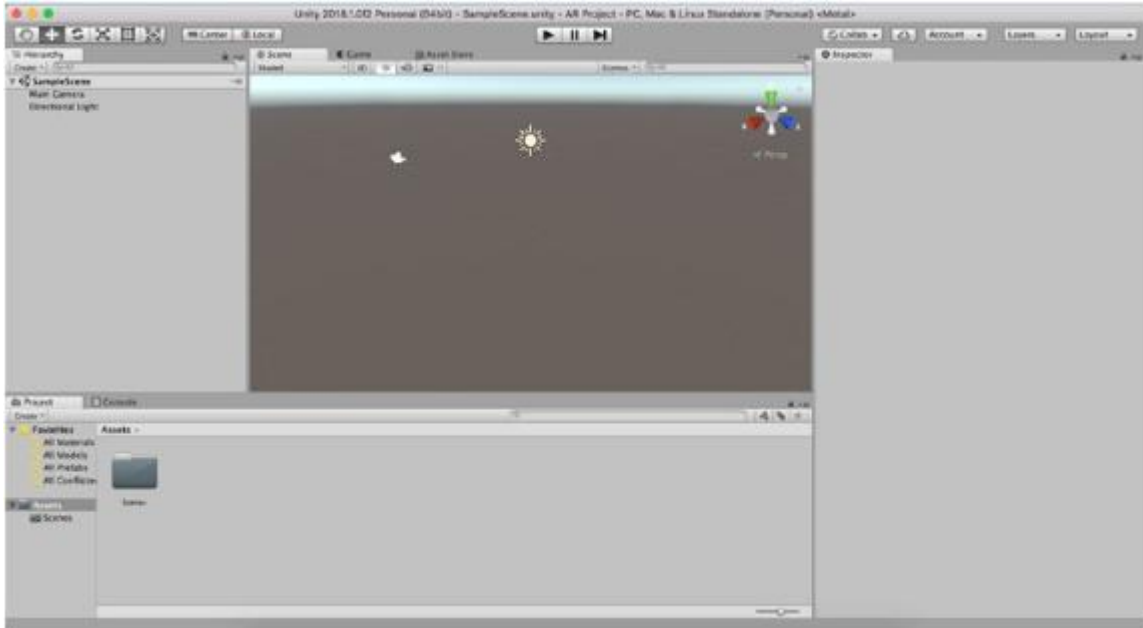
Gambar 2.1 Layar Proyek Unity

Pilih ikon Baru di kanan atas layar. Ini akan membuka layar Proyek Baru Unity (Gambar 2.2). Di kotak dialog, ketik Nama proyek, tetapkan lokasi file Unity, ketik nama organisasi Anda, dan tetapkan Templat ke 3D.



Gambar 2.2 Layar Proyek Baru Unity

Pada kotak dialog layar Unity New Project, masukkan nama proyek, pilih tempat penyimpanan berkas ini, lalu pilih ikon Create project (Gambar 2.2). Saya telah memilih AR project sebagai nama berkas dan memilih folder Users pada hard drive. Sekarang Unity akan terbuka dengan proyek Unity kosong (Gambar 2.3).

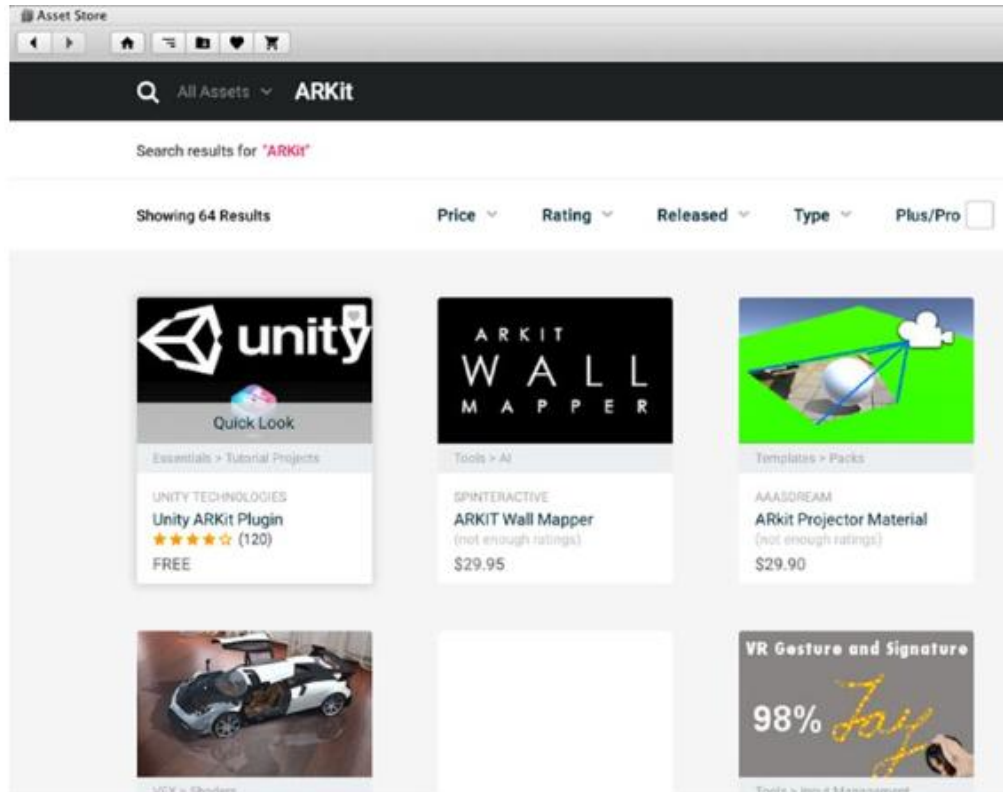


Gambar 2.3 Layar Proyek Unity Yang Kosong

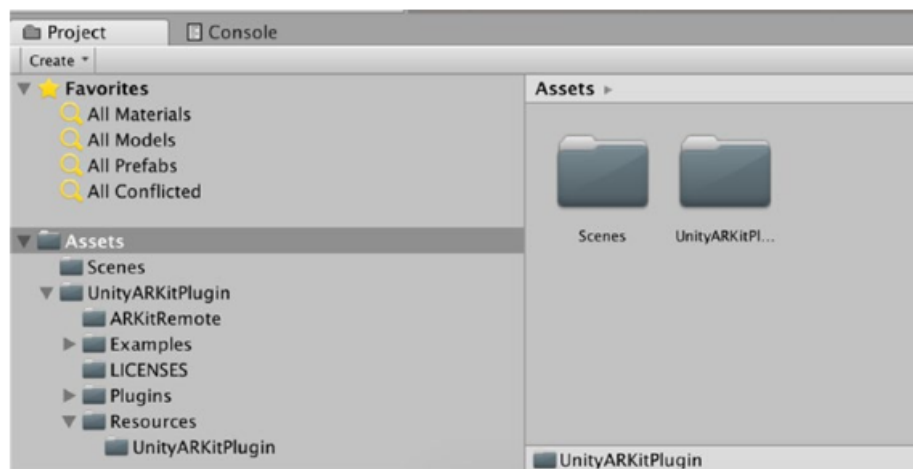
2.2 MEMASANG UNITY ARKIT

Sekarang adalah saat yang tepat untuk memasang Unity ARKit. Untuk memasang Unity ARKit dari Unity Store, Anda perlu mengakses Unity Asset Store. Unity Asset Store dapat diakses dengan sejumlah cara. Di jendela utama layar Unity, terdapat tab Asset Store, dan ini akan memunculkan Jendela Asset Store. Jendela ini juga dapat diakses dengan menggunakan tombol Command dan 9 ($\text{⌘}+9$). Untuk mengunduh sumber daya dari Asset Store, Anda perlu mendaftar ke Unity untuk membuat Unity ID. Setelah Anda membuat Unity ID, di jendela Unity Asset Store, terdapat bilah pencarian. Ketik di bilah pencarian ARKit, dan ini akan memunculkan daftar file yang memenuhi kriteria pencarian ini (Gambar 2.4).

Pengaturan jendela default di Unity akan memperlihatkan jendela Unity Store yang diperkecil; untuk melihat Asset Store dalam mode Layar Penuh, terdapat menu tarik-turun di sisi kanan atas layar. Pilih ini dan klik dengan tombol kiri tetikus. Ini akan menampilkan opsi layar, Muat Ulang, Maksimalkan, Tutup Tab, dan Tambahkan Tab. Pilih opsi maksimalkan (klik tombol kiri tetikus). Di bagian atas layar, terdapat beberapa opsi filter. Di bawah opsi filter, akan terdapat aset yang memenuhi kriteria pencarian Anda. Klik dua kali ARKit, dan ini akan memuat layar untuk aset ini. Pilih tombol impor, dan Aset ini akan diimpor ke Unity (Gambar 2.5).



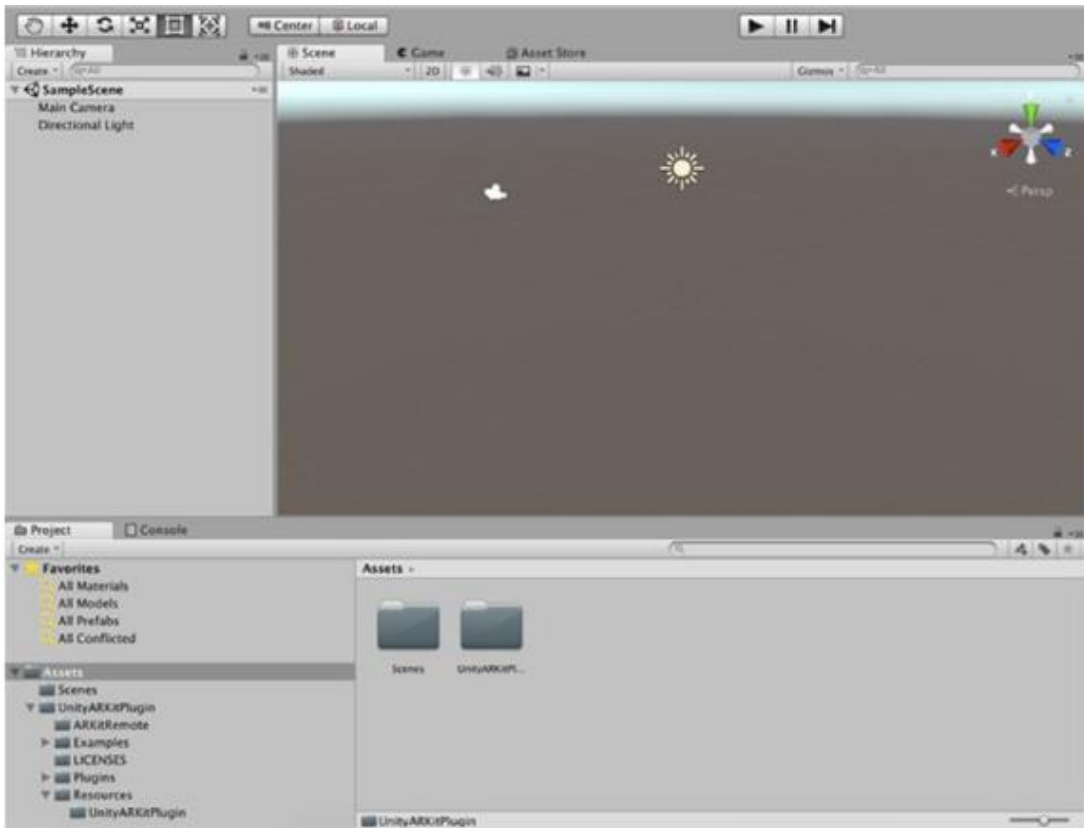
Gambar 2.4 Unity Asset Store



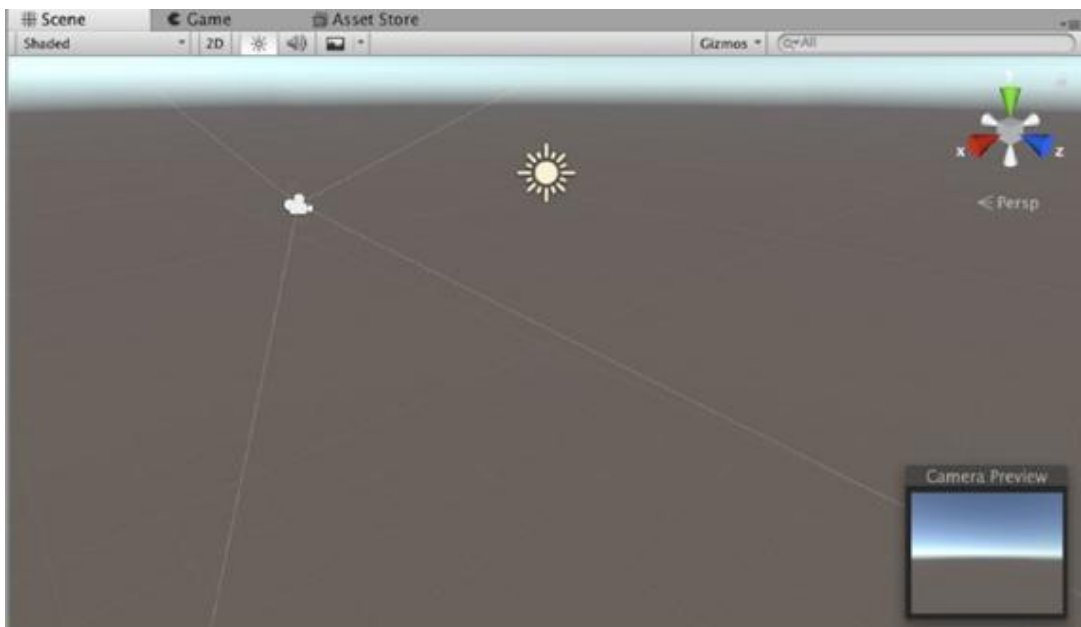
Gambar 2.5 Folder Proyek Unity Dengan Arkit Di Folder Scene

2.3 TATA LETAK EDITOR UNITY

Sekarang saatnya untuk melihat lebih dekat tata letak Editor Unity. Jendela utama dibagi menjadi beberapa panel. Tampilan default yang ditampilkan (Pengaturan pabrik) untuk suatu area dipilih dengan mengklik tab tampilan. Tampilan dapat ditambahkan, dipindahkan, dihapus, dan diubah ukurannya, dan Editor mendukung peralihan di antara tata letak, jadi tata letak pada dasarnya adalah susunan tampilan tertentu.

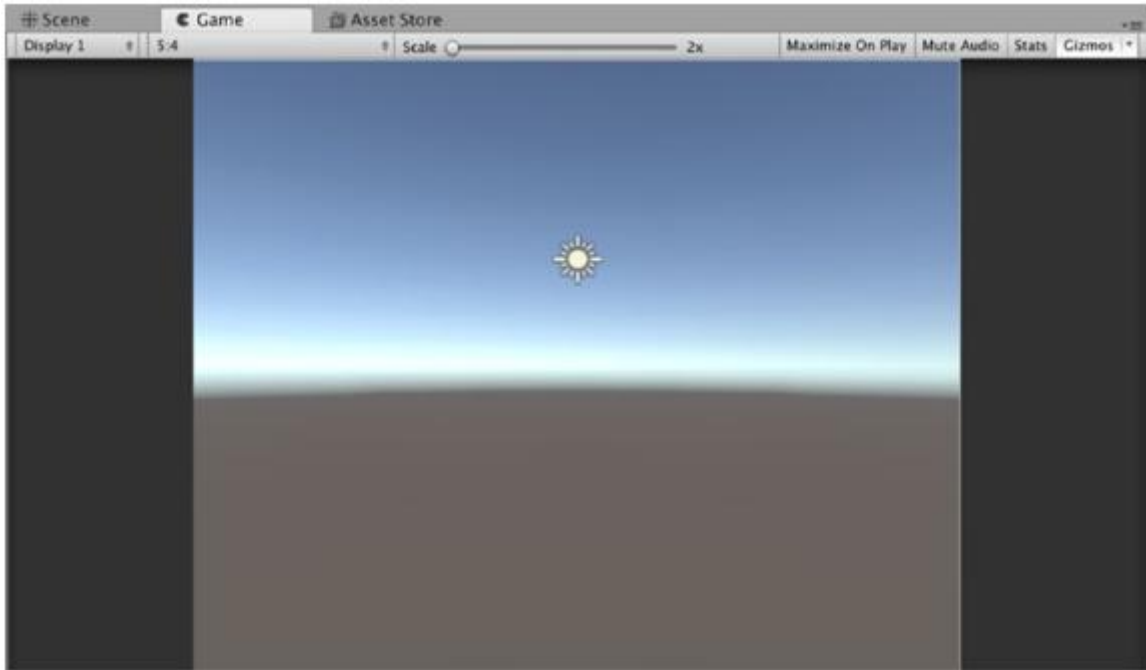


Gambar 2.6 Tata Letak Default Unity Editor



Gambar 2.7 Tampilan Scene Dipilih Di Area Multitab

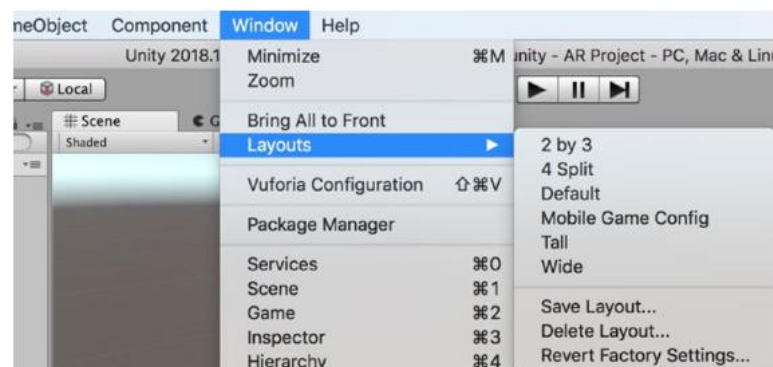
Misalnya, tata letak default jendela utama (Gambar 2.6) memiliki area yang berisi Tampilan Scene (Gambar 2.7) dan Tampilan Game (Gambar 2.8).



Gambar 2.8 Tampilan Game Dipilih Di Area Multitab

Tata Letak Prasetel

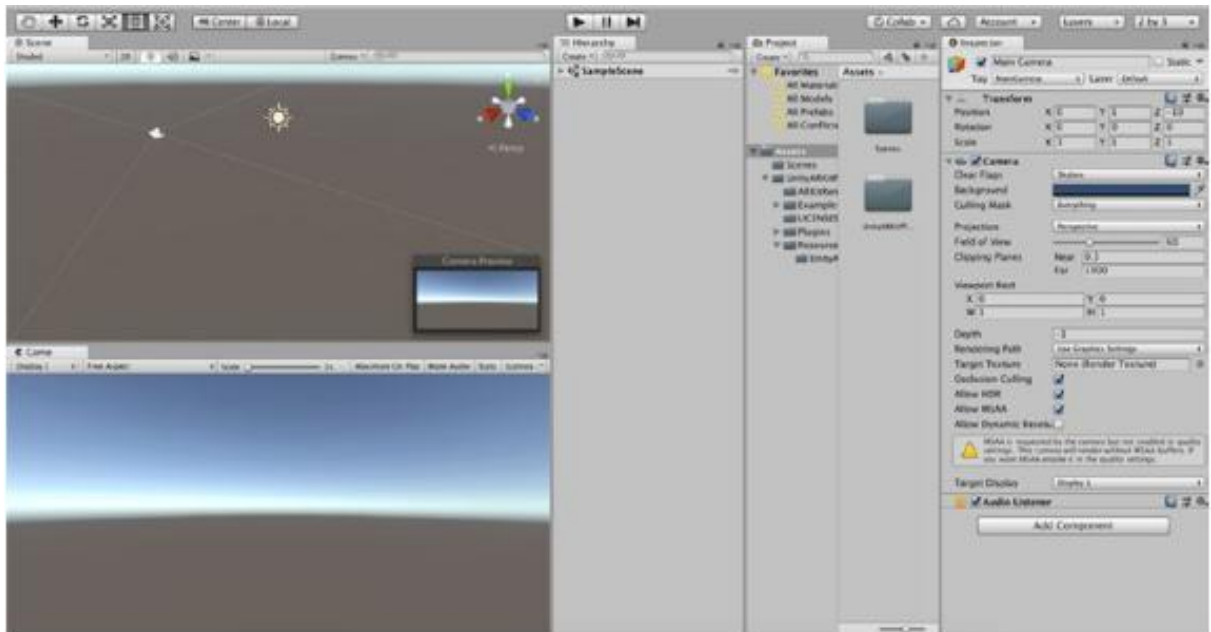
Tata letak default hanyalah salah satu dari beberapa tata letak prasetel. Tata letak alternatif dapat dipilih dari menu di sudut kanan atas jendela utama (Gambar 2-9). Unity juga memungkinkan kita membuat tata letak sendiri. Pada Gambar 2-9, Anda akan melihat bahwa menu saya memiliki Konfigurasi Game Seluler. Ini adalah Tata Letak kustom yang saya buat saat membuat Game Seluler. Lihat berbagai Tata Letak di menu Anda. Gambar 2-6 hingga 2-7 menunjukkan tata letak yang dihasilkan.



Gambar 2.9 Menu Tata Letak

Saya akan menjelaskan masing-masing jenis tampilan secara lebih rinci nanti, tetapi untuk saat ini, perhatikan bahwa tata letak 2x3 (Gambar 2.10) adalah contoh tata letak tempat Tampilan Adegan dan Tampilan Game berada di area terpisah, bukan berbagi di area yang sama. Tata letak 4-split (Gambar 2.11) memiliki empat contoh Tampilan Adegan, yang menunjukkan bahwa tata letak tidak terbatas pada satu dari setiap jenis tampilan. Tata Letak Tinggi (Gambar 2.12) menyediakan Tampilan Adegan Potret. Tata Letak Lebar (Gambar 2.13) menyediakan

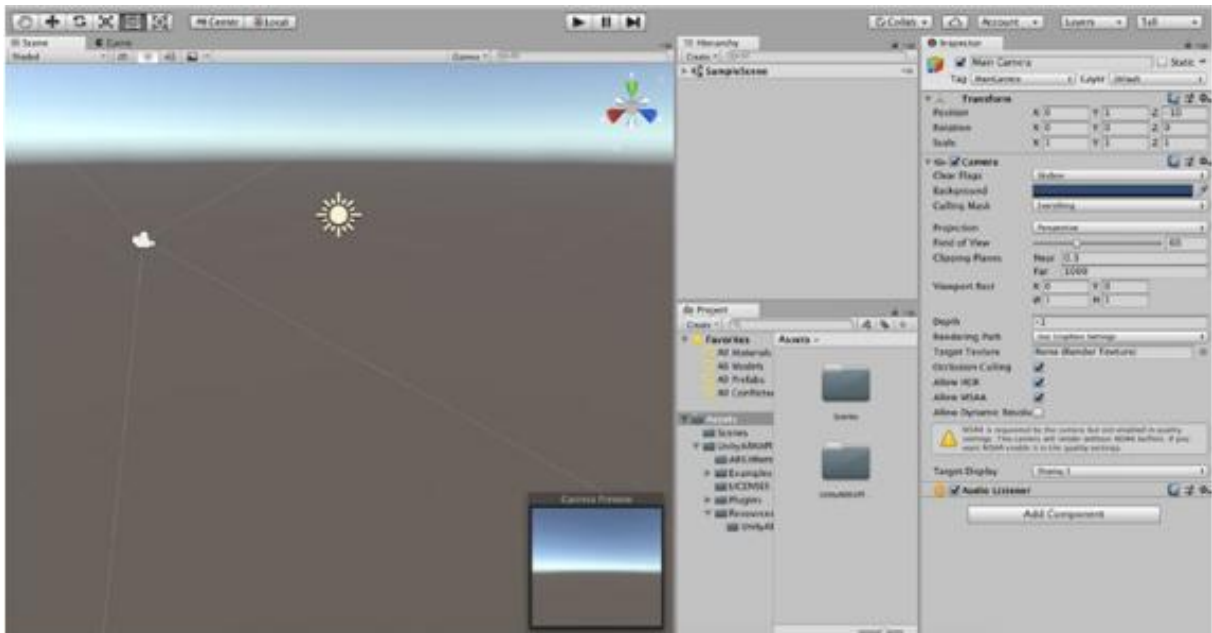
Tampilan Adegan lanskap.



Gambar 2.10 Tata Letak 2x3



Gambar 2.11 Tata Letak 4-Split



Gambar 2.12 Tata Letak Tall



Gambar 2.13 Tata Letak Lebar

Tata Letak Kustom

Tata letak prasetel menyediakan berbagai ruang kerja, tetapi untungnya, Anda tidak dibatasi untuk menggunakannya sebagaimana adanya. Unity menyediakan fleksibilitas untuk mengatur ulang jendela Editor sepenuhnya sesuai keinginan Anda.

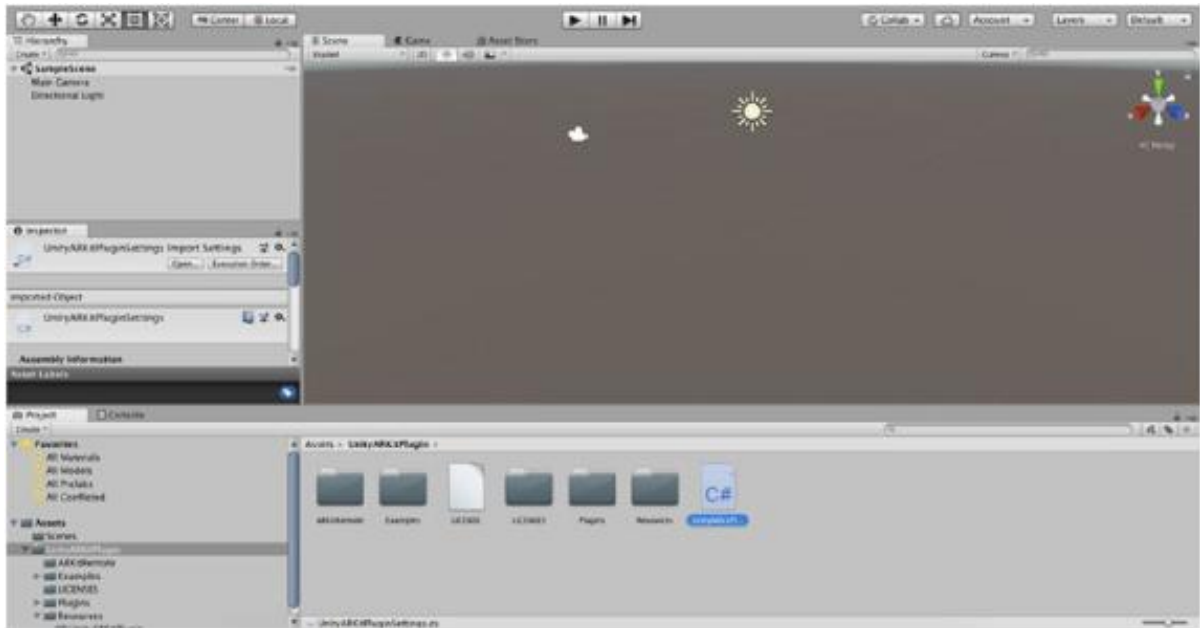
2.4 UBAH UKURAN AREA

Sebagai permulaan, Anda mungkin memperhatikan saat mencoba berbagai tata letak prasetel bahwa beberapa area terlalu sempit, misalnya, di panel kiri tata letak Lebar (Gambar 2.13). Untungnya, Anda dapat mengklik batas area dan menyeretnya untuk mengubah

ukuran area.

Pindahkan Tampilan

Yang lebih keren lagi, Anda dapat memindahkan tampilan. Menyeret tab tampilan ke wilayah tab lain akan memindahkan tampilan ke sana. Dan menyeret tab ke area "docking" akan membuat area baru. Misalnya, mulailah dengan tata letak Default, dan seret tab Inspektur ke kanan tab Hierarki. Sekarang Tampilan Inspektur berbagi area yang sama dengan Tampilan Hierarki. Hasilnya akan terlihat seperti Gambar 2.14.

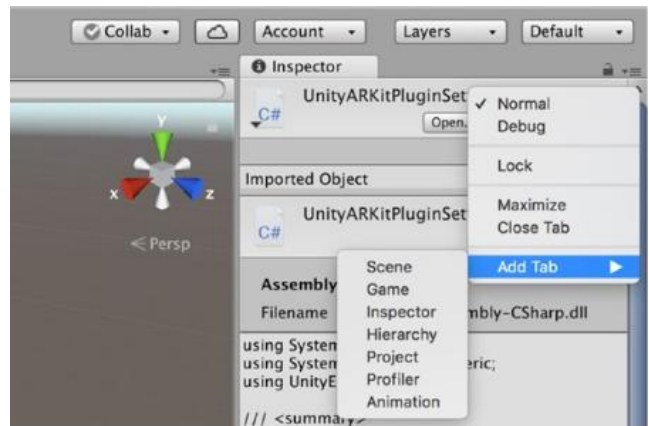


Gambar 2.14 Ruang Kerja Disesuaikan Dengan Tampilan Yang Dipindahkan

Pisahkan Tampilan

Anda bahkan dapat menyeret tampilan ke luar jendela Editor sehingga berada di jendela "mengambang"-nya sendiri, yang dapat diperlakukan seperti area lainnya. Seret tab Adegan ke luar Editor, sehingga berada di jendela mengambang, lalu seret tab Permainan ke dalam wilayah tabnya. Hasilnya akan terlihat seperti Gambar 2.15. Demikian pula, menyeret tab ke wilayah docking jendela mengambang akan menambahkan area lain ke jendela.

Saya suka memisahkan Tampilan Permainan ke dalam jendela mengambang, karena saya biasanya tidak perlu melihatnya saat bekerja di Editor hingga saya mengklik Putar, dan ini memungkinkan saya untuk memaksimalkan Tampilan Permainan hingga memenuhi seluruh layar. Saya juga suka bekerja dengan lebih dari satu monitor. Dengan cara ini, saya dapat memaksimalkan ruang layar saya.



Gambar 2.15 Daftar Tampilan Baru

Jendela mengambang sering kali tertutup oleh jendela lain, jadi menu Windows pada bilah menu memiliki item menu untuk membuat setiap tampilan terlihat. Perhatikan bahwa ada pintasan papan ketik untuk setiap tampilan, dan ada juga submenu Tata Letak yang identik dengan menu tata letak di dalam Editor.

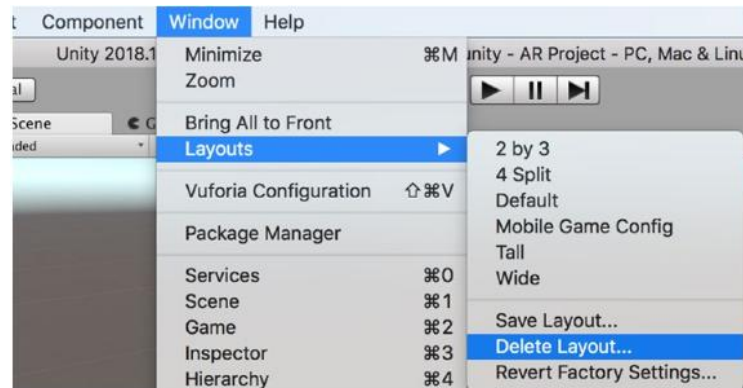
Tambah dan Hapus Tampilan

Anda juga dapat menambah dan menghapus tampilan di setiap area menggunakan menu di sudut kanan atas area (Gambar 2.16). Item Tutup Tab menghapus tampilan yang sedang ditampilkan. Item Tambah Tab menyediakan daftar tampilan baru untuk Anda pilih. Anda mungkin ingin memiliki tata letak yang berbeda untuk platform target yang berbeda, atau tata letak yang berbeda untuk pengembangan vs. pengujian permainan, atau bahkan tata letak yang berbeda untuk permainan yang berbeda. Misalnya, saya memiliki tata letak khusus untuk permainan seluler saya yang menyimpan Tampilan Permainan dalam rasio aspek potret yang sesuai. Akan merepotkan untuk mengonfigurasi ulang Editor secara manual setiap kali Anda memulai Unity. Untungnya, Anda dapat memberi nama dan menyimpan tata letak dengan memilih opsi Simpan Tata Letak di menu tata letak, yang akan meminta Anda memasukkan nama tata letak baru (Gambar 2.16).

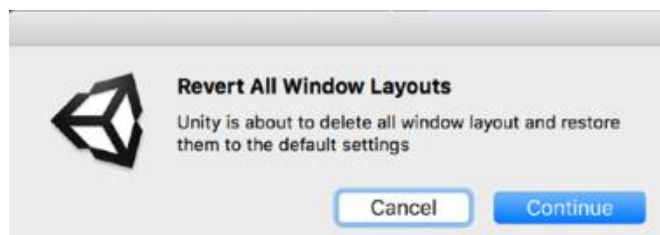


Gambar 2.16 Perintah Untuk Tata Letak Baru

Setelah disimpan, tata letak baru akan tercantum dalam menu tata letak dan juga dalam daftar tata letak yang tersedia untuk dihapus jika Anda memilih Hapus Tata Letak (Gambar 2.17).



Gambar 2.17 Menu Penghapusan Untuk Tata Letak

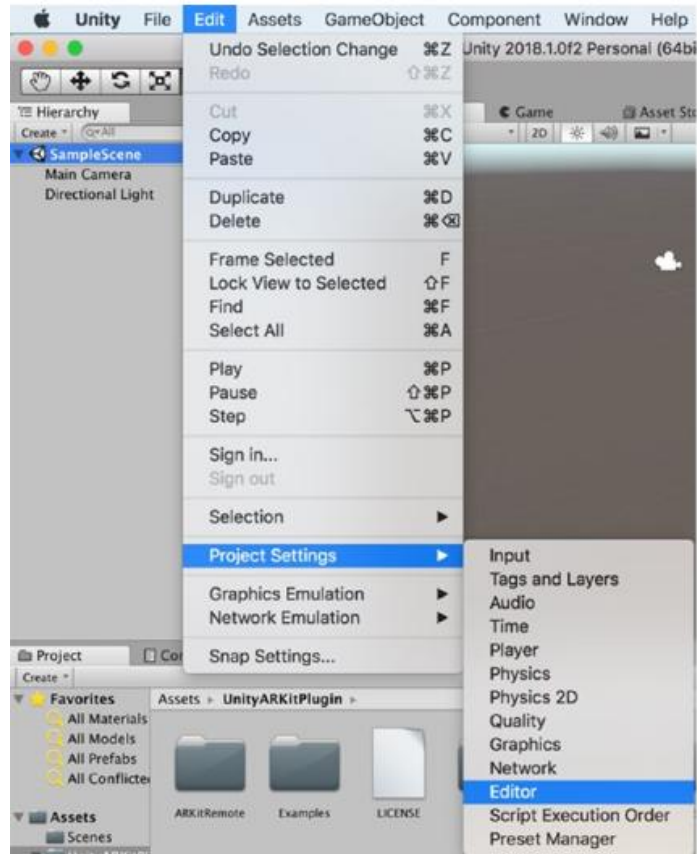


Gambar 2.18 Mengembalikan Pengaturan Tata Letak Asli

Jika Anda mengacaukan atau menghapus tata letak asli, Anda dapat memilih opsi Restore Factory Settings di menu area (Gambar 2.18). Ini juga akan menghapus semua tata letak kustom. Jika Anda mengubah tata letak dan belum menyimpan perubahan, Anda selalu dapat membuangnya dengan memilih ulang tata letak tersebut di menu tata letak.

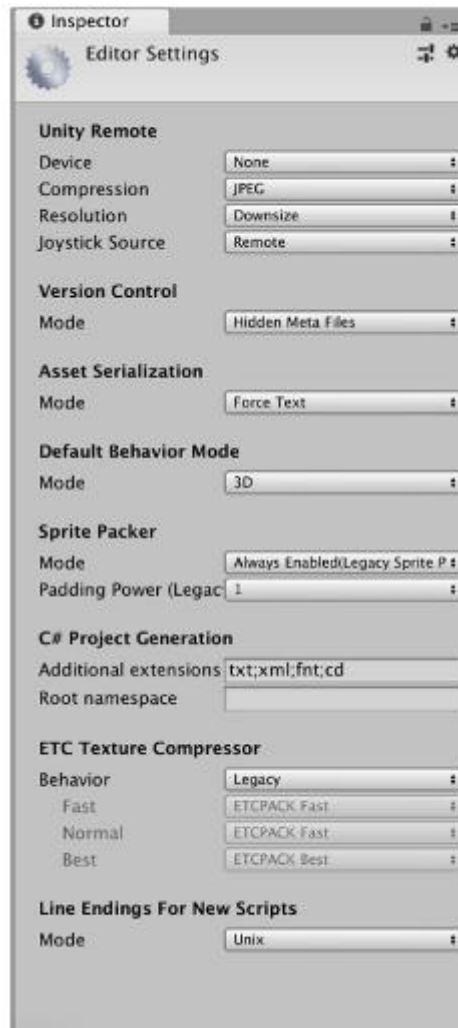
Tampilan Inspektur

Tampilan terbaik yang dijelaskan secara terperinci pertama kali adalah Tampilan Inspektur karena fungsinya adalah untuk menampilkan informasi tentang objek yang dipilih di tampilan lain. Tampilan ini lebih dari sekadar inspektur karena biasanya dapat digunakan untuk mengubah item yang dipilih. Tampilan Inspektur juga digunakan untuk menampilkan dan menyesuaikan berbagai pengaturan yang dapat ditampilkan di menu Edit (Gambar 2.19).



Gambar 2.19 Menampilkan Pengaturan Editor

Tampilan Inspektur menampilkan Pengaturan Editor. Jika proyek saat ini memiliki metafile, maka Mode Kontrol Versi diatur ke Meta File (dan jika Anda menggunakan Server Aset, opsi ini diatur ke Server Aset). Untuk menyembunyikan metafile, atur Mode Kontrol Versi ke Tersembunyi (Gambar 2.20).



Gambar 2.20 Pengaturan Editor Di Tampilan Inspektur

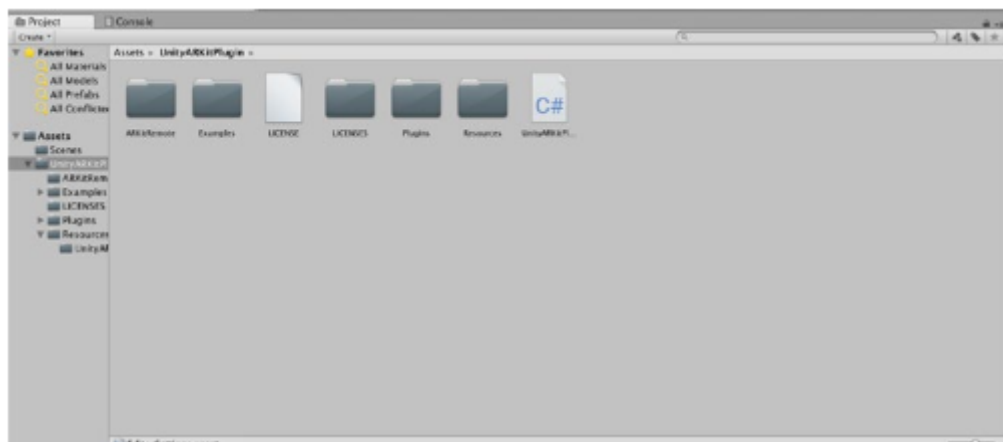
Dengan Mode Kontrol Versi yang ditetapkan ke Dinonaktifkan, Unity akan menghapus metafile. Pelacakan aset kini ditangani dalam file biner di dalam folder Pustaka proyek. Pengguna Unity yang menggunakan metafile untuk dukungan kontrol versi juga memiliki opsi untuk menetapkan Mode Serialisasi Aset ke Paksa Teks. Dalam mode tersebut, file adegan Unity disimpan dalam format YAML (YAML Bukan Bahasa Markup) teks saja. Biasanya, Tampilan Inspektur menampilkan properti objek yang terakhir dipilih (ketika Anda membuka Pengaturan Editor, Anda benar-benar memilihnya). Namun terkadang Anda tidak ingin Tampilan Inspektur berubah saat Anda memilih objek lain. Dalam kasus tersebut, Anda dapat menyematkan Tampilan Inspektur ke objek dengan memilih opsi Kunci di menu di kanan atas tampilan (Gambar 2.21).



Gambar 2.21 Mengunci Tampilan Inspektur

Tampilan Proyek

Meskipun Tampilan Inspektur dapat dianggap sebagai Tampilan tingkat terendah di Editor, karena hanya menampilkan properti dari satu objek, Tampilan Proyek dapat dianggap sebagai tampilan tingkat tertinggi (Gambar 2.22). Tampilan Proyek menampilkan semua aset yang tersedia untuk game Anda, mulai dari model, tekstur, dan skrip individual hingga file adegan yang menyertakan aset tersebut. Semua aset proyek adalah file yang berada di folder Aset proyek Anda (jadi Anda mungkin ingin menganggap Tampilan Proyek sebagai Tampilan Aset).



Gambar 2.22 Tingkat Atas Tampilan Proyek

Beralih Antara Satu Kolom dan Dua Kolom. Pada beberapa versi Unity yang lebih lama, Tampilan Proyek hanya memiliki tampilan satu kolom. Opsi tersebut masih tersedia di menu

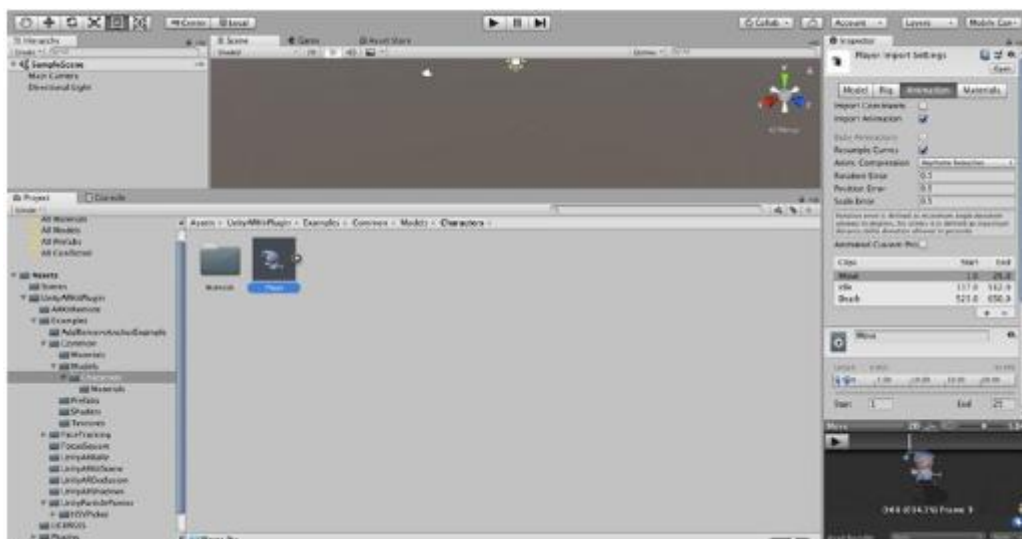
untuk Tampilan Proyek (klik ikon tiga garis kecil di kanan atas tampilan), jadi Anda sekarang dapat beralih antara satu dan dua kolom.

2.5 SKALA IKON

Penggeser di bagian bawah mengubah skala tampilan di panel kanan—skala yang lebih besar bagus untuk tekstur, dan yang lebih kecil lebih baik untuk item seperti skrip yang tidak memiliki ikon yang menarik. Ini adalah alasan yang bagus untuk mempartisi aset berdasarkan jenis aset (misalnya, letakkan semua tekstur dalam folder Tekstur, skrip dalam folder Skrip, dan seterusnya). Kemungkinannya, pengaturan penggeser skala tunggal tidak akan bagus untuk campuran jenis aset.

Periksa Aset

Memilih aset di sebelah kanan akan menampilkan properti aset tersebut di Tampilan Inspektur. Misalnya, jika Anda memilih contoh animasi, Tampilan Inspektur akan menampilkan informasi tentang animasi tersebut, yang sebagian dapat Anda ubah, seperti durasinya, dan bahkan memungkinkan Anda memutar animasi di Editor (Gambar 2.23). Kita akan membahas cara mengubah properti aset di bab berikutnya, tetapi untuk saat ini, silakan pilih berbagai jenis aset di Tampilan Proyek dan lihat apa yang muncul di Tampilan Inspektur.



Gambar 2.23 Memeriksa Aset Yang Dipilih Dalam Tampilan Proyek

Mencari Aset

Dalam proyek yang besar dan kompleks, sulit untuk mencari aset tertentu secara manual. Untungnya, seperti di Finder, ada kotak pencarian yang dapat digunakan untuk memfilter hasil yang ditampilkan di panel kanan tampilan Proyek. Pada Gambar 2.24, Tampilan Proyek menampilkan hasil pencarian aset dengan kata "tambah" pada namanya.



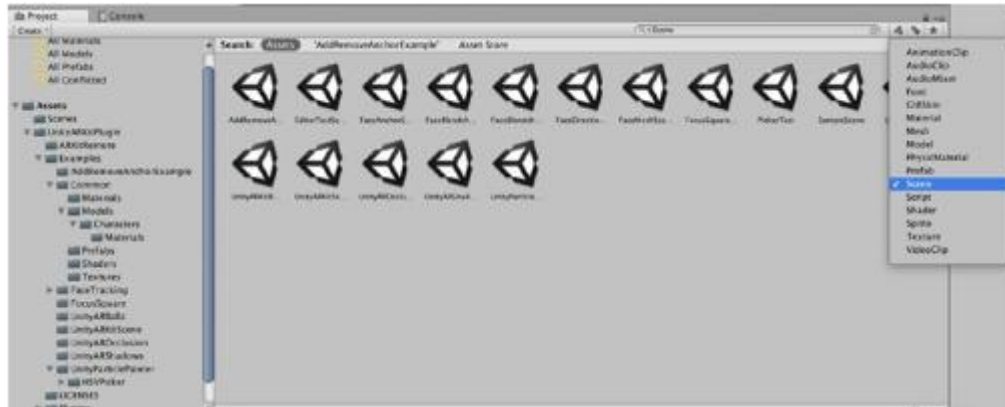
Gambar 2.24 Mencari Aset Dengan Kata "Add" Pada Namanya

Panel kanan menampilkan hasil pencarian untuk semua aset di bawah Aset (yaitu, semua aset kita). Pencarian dapat dipersempit lebih jauh dengan memilih salah satu subfolder di panel kiri. Misalnya, jika Anda tahu bahwa Anda sedang mencari sebuah adegan, dan Anda telah mengatur aset Anda ke dalam subfolder berdasarkan jenis aset, Anda dapat memilih folder untuk mencari. Pada Gambar 2.25, saya telah mencari aset apa pun dengan kata add pada nama file di folder contoh.



Gambar 2.25 Mencari Aset Dalam Folder Tertentu

Perhatikan tepat di bawah pencarian; ada tab dengan nama folder yang dipilih. Anda masih dapat mengklik tab Aset di sebelah kiri untuk melihat hasil pencarian untuk semua aset Anda, baik secara lokal maupun di Unity Asset Store, yang akan kami gunakan dalam banyak hal dalam buku ini. Anda juga dapat memfilter pencarian menurut jenis aset, menggunakan menu tepat di sebelah kanan kotak pencarian. Daripada hanya mencari di folder Contoh, Anda dapat memilih adegan sebagai jenis aset yang diinginkan (Gambar 2.26). Perhatikan bagaimana hal itu mengakibatkan penambahan ke kotak pencarian. Awalan t: menunjukkan pencarian harus difilter menurut jenis aset berikut. Anda dapat mengetiknya tanpa menggunakan menu.



Gambar 2.26 Pencarian yang difilter menurut jenis aset

Tombol di sebelah kanan menu jenis aset adalah untuk memfilter menurut label (Anda dapat menetapkan label untuk setiap aset di Tampilan Inspektur), yang juga cukup berguna untuk mencari di Asset Store. Dan tombol paling kanan, bintang, akan menyimpan pencarian saat ini di bagian Favorit di panel kiri.

Operasikan Aset

Aset di Tampilan Proyek dapat dimanipulasi seperti file terkaitnya di Finder. Mengklik dua kali aset akan mencoba membuka program yang sesuai untuk melihat atau mengedit aset tersebut. Ini sama saja dengan mengklik kanan aset dan memilih Buka. Mengklik dua kali file adegan akan membuka adegan tersebut di jendela Editor Unity ini, sama seperti jika Anda memilih Buka Adegan di menu File.

Anda juga dapat mengganti nama, menduplikasi dan menghapus, serta menyeret file ke dalam dan ke luar folder sama seperti yang dapat Anda lakukan di Finder. Beberapa operasi tersedia di menu Edit Unity dan di menu pop-up saat Anda mengklik kanan aset. Anda akan berlatih dengan itu di beberapa bab berikutnya.

Demikian pula, di bab berikutnya, Anda akan mengerjakan penambahan aset ke proyek. Itu melibatkan pengimporan file atau pengimporan paket Unity, menggunakan menu Aset di bilah menu atau cukup menyeret file ke folder Aset proyek menggunakan Finder.

Tampilan Hirarki

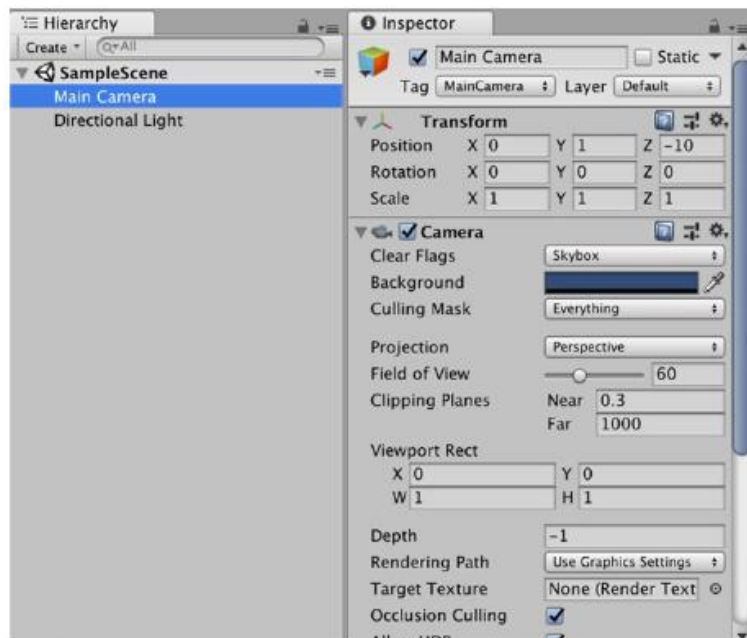
Setiap mesin gim memiliki objek tingkat atas yang disebut objek atau entitas gim untuk mewakili apa pun yang memiliki posisi, perilaku potensial, dan nama untuk mengidentifikasinya. Objek game Unity adalah contoh dari kelas GameObject. Secara umum, saat kita merujuk ke jenis objek Unity, kita akan menggunakan nama kelasnya agar tepat dan menjelaskan bagaimana objek tersebut akan dirujuk dalam skrip. Tampilan Hirarki adalah representasi lain dari adegan saat ini.

Sementara Tampilan Adegan adalah representasi 3D dari adegan yang dapat Anda gunakan seperti yang Anda lakukan dengan alat pembuatan konten, dan Tampilan Game menunjukkan adegan seperti yang terlihat saat memainkan game, Tampilan Hirarki mencantumkan semua GameObject dalam adegan dalam struktur pohon yang mudah dinavigasi.

Periksa Objek Game

Saat Anda mengklik GameObject di Tampilan Hierarki, objek tersebut menjadi pilihan Editor saat ini, dan komponen-komponennya ditampilkan di Editor. Setiap GameObject memiliki Komponen Transformasi, yang menentukan posisi, rotasi, dan skalanya, relatif terhadap induknya dalam hierarki (jika Anda memahami matematika grafik 3D, Transformasi pada dasarnya adalah matriks transformasi objek). Beberapa komponen menyediakan fungsi untuk objek game (misalnya, lampu adalah GameObject dengan Komponen Lampu yang terpasang).

Komponen lain merujuk aset seperti jaring, tekstur, dan skrip. Gambar 2.27 menunjukkan komponen GameObject Kamera Utama (dalam tampilan Hierarki, seluruh pohon Pemain GameObject ditampilkan dalam warna biru karena ditautkan ke prefab, jenis aset khusus yang digunakan untuk mengkloning GameObject atau sekelompok GameObject).



Gambar 2.27 Tampilan Hirarki Dan Tampilan Inspektur

2.6 GAMEOBJECT INDUK DAN ANAK

Anda akan menemukan bahwa banyak GameObject disusun dalam hierarki, oleh karena itu nama tampilan ini. Pengasuhan masuk akal untuk objek game yang secara konseptual dikelompokkan bersama. Misalnya, saat Anda ingin menggerakkan mobil, Anda ingin rodanya bergerak otomatis bersama mobil. Jadi, roda harus ditetapkan sebagai anak mobil, yang berjarak dari bagian tengah mobil. Saat roda berputar, roda berputar relatif terhadap pergerakan mobil. Pengasuhan juga memungkinkan kita untuk mengaktifkan atau menonaktifkan seluruh kelompok objek game sekaligus.

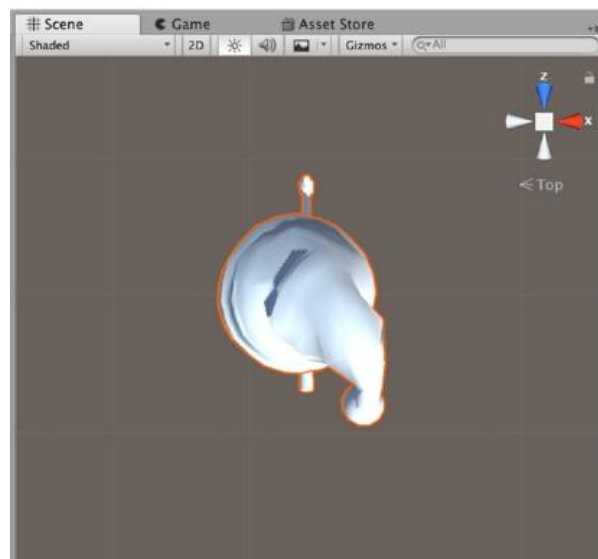
Tampilan Adegan

Sementara Tampilan Hirarki memungkinkan kita untuk membuat, memeriksa, dan memodifikasi GameObject dalam adegan saat ini, tampilan ini tidak memberi kita cara untuk memvisualisasikan adegan tersebut. Di situlah Tampilan Adegan berperan. Tampilan Adegan

mirip dengan antarmuka aplikasi pemodelan 3D. Ini memungkinkan Anda memeriksa dan memodifikasi pemandangan dari sudut pandang 3D mana pun dan memberi Anda gambaran tentang bagaimana produk akhir akan terlihat.

Menavigasi Pemandangan

Jika Anda tidak terbiasa bekerja di ruang 3D, ini adalah perluasan langsung dari bekerja di 2D. Alih-alih hanya bekerja di ruang dengan sumbu x dan y dan koordinat (x,y), dalam ruang 3D, Anda memiliki sumbu z tambahan dan koordinat (x,y,z). Sumbu x dan z menentukan bidang dasar, dan y mengarah ke atas (Anda dapat menganggap y sebagai tinggi).

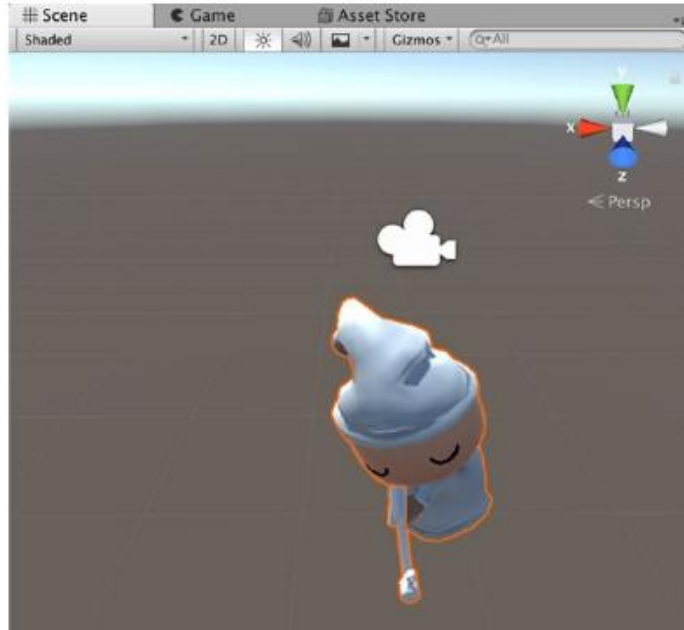


Gambar 2.28 Tampilan Atas Dalam Scene View

Beberapa aplikasi 3D dan mesin permainan menggunakan sumbu z untuk tinggi dan sumbu x dan y untuk bidang dasar, jadi saat mengimpor aset, Anda mungkin harus menyesuaikan (memutarkannya). Sudut pandang dalam ruang 3D biasanya disebut kamera. Meneklik tanda panah x, y, dan z pada Scene Gizmo warna-warni di sudut kanan atas adalah cara cepat untuk membalik kamera sehingga menghadap sepanjang sumbu masing-masing. Misalnya, mengklik tanda panah y akan memberikan Anda tampilan atas-bawah dari scene (Gambar 2.28), dan teks di bawah Scene Gizmo bertuliskan “Atas”.

Kamera di sini tidak sama dengan GameObject Kamera dalam scene yang digunakan selama permainan, jadi Anda tidak perlu khawatir mengacaukan permainan saat melihat-lihat di Scene View. Untuk menunjukkan cara menggunakan alat Navigasi, saya telah memilih GameObject Pemain dari Folder Proyek dan menyeretnya ke Hierarchy View. Meneklik kotak di tengah Scene Gizmo akan mengubah proyeksi kamera antara perspektif, yang membuat objek lebih kecil saat objek tersebut menjauh di kejauhan; dan ortografis, yang membuat semua objek pada ukuran aslinya, baik dari dekat maupun jauh.

Perspektif lebih realistis, dan biasanya digunakan dalam permainan, tetapi ortografis sering kali lebih praktis saat mendesain (karenanya ada di mana-mana dalam aplikasi desain berbantuan komputer). Grafik kecil sebelum teks di bawah Scene Gizmo menunjukkan proyeksi saat ini.



Gambar 2.29 Perspektif Miring Dalam Tampilan Adegan

Anda dapat memperbesar dan memperkecil tampilan menggunakan roda gulir tetikus atau dengan memilih alat Tangan di bilah alat kanan atas jendela Editor dan mengeklik-menyeret tetikus sambil menahan tombol Kontrol. Saat alat Tangan dipilih, Anda juga dapat menggerakkan kamera dengan mengeklik-menyeret tampilan, dan Anda dapat memutar (mengorbit) kamera dengan menyeret tetikus sambil menahan tombol Opsi (atau Alt), jadi Anda tidak terbatas pada sudut kamera sumbu saja, seperti pada Gambar 2.29.

Perhatikan bahwa ketika Anda melihat dari sudut sembarangan, teks di bawah Scene Gizmo akan bertuliskan Persp atau Iso, tergantung apakah Anda menggunakan perspektif atau proyeksi ortografis (Iso adalah kependekan dari isometrik, yang merupakan tampilan ortografis miring yang umum dalam gim seperti StarCraft). Tombol lain pada bilah alat mengaktifkan mode untuk menggerakkan, memutar, dan menskalakan GameObjects. Tidak ada alasan untuk mengubahnya saat ini, jadi mode tersebut akan dijelaskan lebih rinci saat Anda mulai membuat proyek baru. Tips Jika Anda tidak sengaja membuat perubahan pada adegan, Anda dapat memilih Batalkan dari menu Edit.

Jika Anda membuat banyak perubahan yang tidak ingin disimpan, Anda dapat menolak untuk menyimpan adegan ini saat beralih ke adegan lain atau keluar dari Unity. Sementara itu, perhatikan bahwa Anda masih dapat menggerakkan kamera saat dalam mode tersebut, menggunakan kombinasi keyboard dan mouse alternatif. Tabel 2.1 mencantumkan semua opsi yang memungkinkan.

Tabel 2.1 Kontrol Kamera Tampilan Adegan yang Tersedia

Aksi	Alat Tangan	Mouse 1 Tombol atau Trackpad	Mouse 2 Tombol	Mouse 3 Tombol
Pindah	Klik-drag	Tahan alt-	Tahan alt-	Tahan alt dan

		command dan klik-drag	control dan klik-drag	klik-drag tombol tengah
Orbit	Tahan alt dan klik-drag	Tahan alt dan klik-drag	Tahan alt dan klik-drag	Tahan alt dan klik-drag
Zoom	Tahan control dan klik-drag	Tahan control dan klik-drag atau geser dua jari	Tahan alt dan klik-drag tombol kanan	Tahan alt dan klik-drag tombol kanan atau roda gulir

Ada beberapa fitur navigasi adegan berbasis keyboard yang praktis lainnya. Menekan tombol Panah akan menggerakkan kamera maju, mundur, kiri, dan kanan sepanjang bidang $x-z$ (bidang tanah). Dan menahan tombol kanan tetikus memungkinkan navigasi adegan seperti dalam permainan orang pertama. Tombol AWSD bergerak ke kiri, maju, kanan, dan mundur, masing-masing, dan menggerakkan tetikus mengontrol ke mana kamera (sudut pandang) melihat.

Saat Anda ingin melihat GameObject tertentu di Scene View, terkadang cara tercepat untuk melakukannya adalah dengan memilih GameObject di tampilan Hierarchy, lalu gunakan item menu Frame Selected di menu Edit (perhatikan tombol pintas praktis F). Pada Gambar 2-28, saya mengklik sumbu x dari Scene Gizmo untuk mendapatkan tampilan horizontal, lalu memilih GameObject Pemain di Hierarchy View, dan menekan tombol F (pintasan untuk Frame Selected di menu Edit) untuk memperbesar dan memusatkan pemain di Scene View. Bahasa Indonesia: Anda juga dapat memilih GameObject langsung di Scene View, tetapi Anda harus keluar dari alat Hand terlebih dahulu.

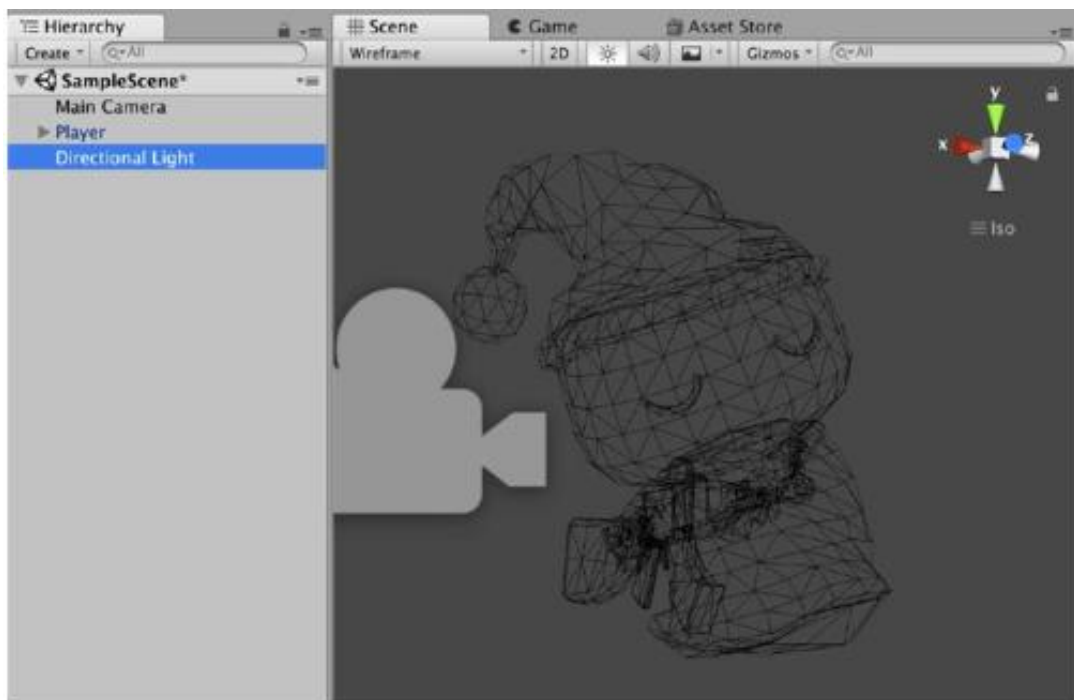
Sama seperti memilih GameObject di Hierarchy View akan menghasilkan pilihan itu ditampilkan di Scene View dan Inspector View, memilih GameObject di Scene View juga akan menampilkan pilihan itu di Inspector View dan menampilkannya sebagai GameObject yang dipilih kembali di Hierarchy view. Pada Gambar 2.30, setelah saya memanggil Frame Selected pada Player, saya mengklik alat Move (tombol tepat di sebelah kanan tombol alat Hand di sudut kanan atas jendela Editor) lalu mengklik GameObject di dekat Player di Scene View. Hierarchy View secara otomatis memperbarui untuk menunjukkan bahwa GameObject dipilih, dan GameObject juga ditampilkan di Inspector View.



Gambar 2.30 Memilih Gameobject Dalam Scene View

2.7 OPSI SCENE VIEW

Tombol-tombol yang berjejer di bagian atas Scene View menyediakan opsi tampilan untuk membantu pengembangan game Anda. Setiap tombol mengonfigurasi mode tampilan. Tombol paling kiri mengatur mode Draw. Biasanya, mode ini diatur ke Textured, tetapi jika Anda ingin melihat semua poligon, Anda dapat mengaturnya ke Wireframe (Gambar 2.31).



Gambar 2.31 Tampilan Wireframe Dalam Tampilan Scene

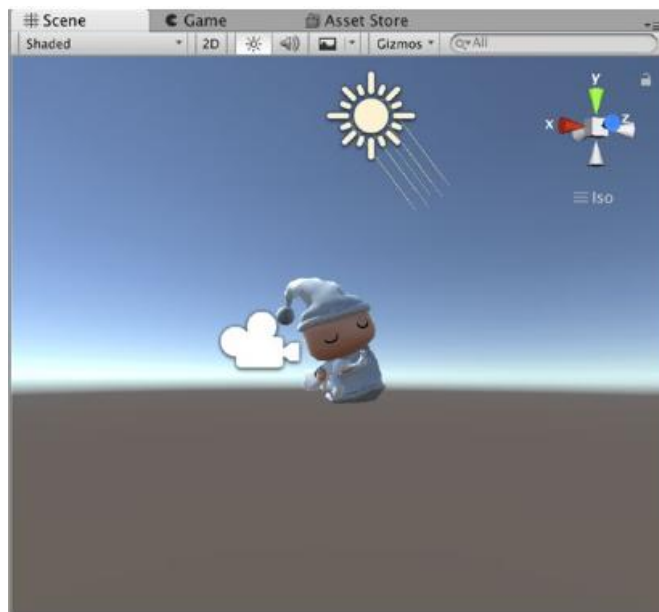
Tombol berikutnya mengatur Render Paths, yang mengontrol apakah scene diwarnai secara
Pengembangan Game AR (Augmented Reality) – Dr. Mars Caroline Wibowo

normal atau untuk diagnostik. Tiga tombol di sebelah kanan tombol mode Render Paths adalah tombol sakelar sederhana. Masing-masing tombol memunculkan beberapa dokumentasi mouse-over (atau dikenal sebagai tooltips) saat Anda mengarahkan mouse ke atasnya. Yang pertama dari tombol-tombol tersebut mengontrol mode Scene Lighting.

Ini beralih antara menggunakan skema pencahayaan default dalam Scene View atau lampu sebenarnya yang telah Anda tempatkan dalam game. Tombol tengah mengaktifkan mode Game Overlay, apakah langit, lens flare, dan efek kabut terlihat. Dan terakhir, ada Mode Audition, yang mengaktifkan dan menonaktifkan suara.

Scene View Gizmos

Tombol Gizmos di sebelah kanan mengaktifkan tampilan grafik diagnostik yang terkait dengan Komponen. Scene View pada Gambar 2.32 memperlihatkan beberapa gizmos. Dengan mengklik tombol Gizmos dan memeriksa daftar gizmos yang tersedia, Anda dapat melihat ikon-ikon yang mewakili Kamera dan Lampu.



Gambar 2.32 Gizmos Dalam Tampilan Adegan

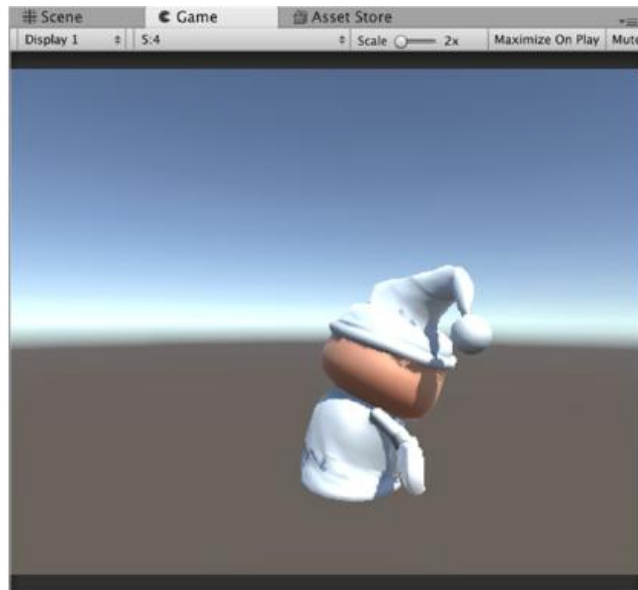
Anda dapat memilih dan membatalkan pilihan berbagai kotak centang di jendela Gizmos untuk fokus pada objek yang Anda minati. Kotak centang di kiri atas beralih antara tampilan 3D dari gizmos atau hanya ikon 2D. Slider yang berdekatan mengontrol skala gizmos (jadi cara cepat untuk menyembunyikan semua gizmos adalah dengan menyeret slider skala ke paling kiri).

2.8 TAMPILAN GAME

Sekarang mari kita lihat Tampilan Game. Seperti Tampilan Hirarki dan Tampilan Adegan, Tampilan Game menggambarkan adegan saat ini, tetapi bukan untuk tujuan pengeditan. Sebaliknya, Tampilan Game ditujukan untuk memainkan dan men-debug game. Tampilan Game muncul secara otomatis saat Anda mengklik tombol Putar di bagian atas jendela Unity Editor. Jika tidak ada Tampilan Game saat Anda mengklik Putar, yang baru akan dibuat. Jika

tampilan Game terlihat saat Editor tidak dalam mode Putar, tampilan tersebut menunjukkan game dalam keadaan awalnya (yaitu, dari sudut pandang posisi Kamera awal).

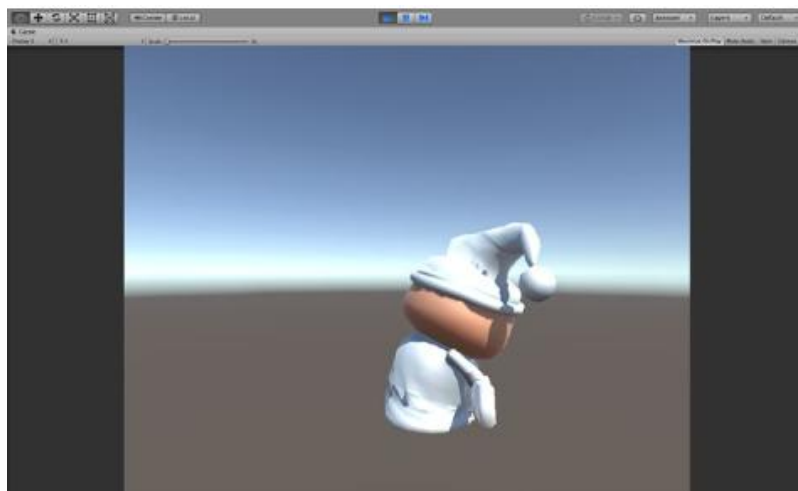
Tampilan Game menunjukkan bagaimana tampilan dan fungsi game saat Anda benar-benar menggunakannya, tetapi mungkin ada perbedaan dari tampilan dan perilakunya pada target build akhir. Salah satu perbedaan yang mungkin adalah ukuran dan rasio aspek Tampilan Game. Ini dapat diubah menggunakan menu di kiri atas tampilan. Gambar 2.33 menunjukkan apa yang terjadi saat Anda beralih dari rasio Aspek Bebas, yang menyesuaikan dengan dimensi tampilan, ke rasio aspek 5:4, yang mengakibatkan tampilan game diperkecil sehingga sesuai dengan area dan mempertahankan rasio aspek yang dipilih.



Gambar 2.33 Tampilan Game

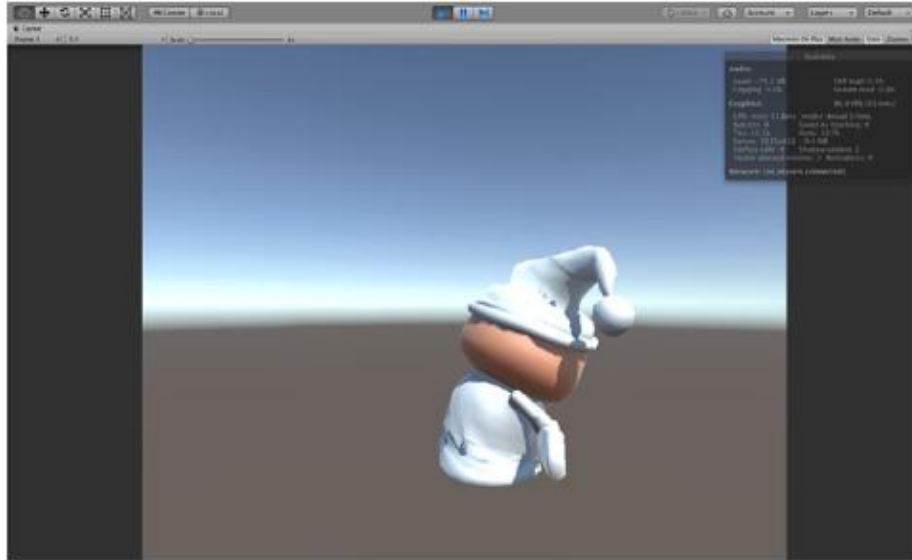
Maksimumkan saat Bermain

Mengklik tombol Maksimumkan saat Bermain akan mengakibatkan tampilan Game meluas hingga memenuhi seluruh jendela Editor saat berada dalam mode Bermain (Gambar 2.34). Jika tampilan dipisahkan dari jendela Editor, tombol tersebut tidak akan berpengaruh.



Gambar 2.34 Tampilan Permainan Dengan Maksimalkan Saat Bermain Statistik

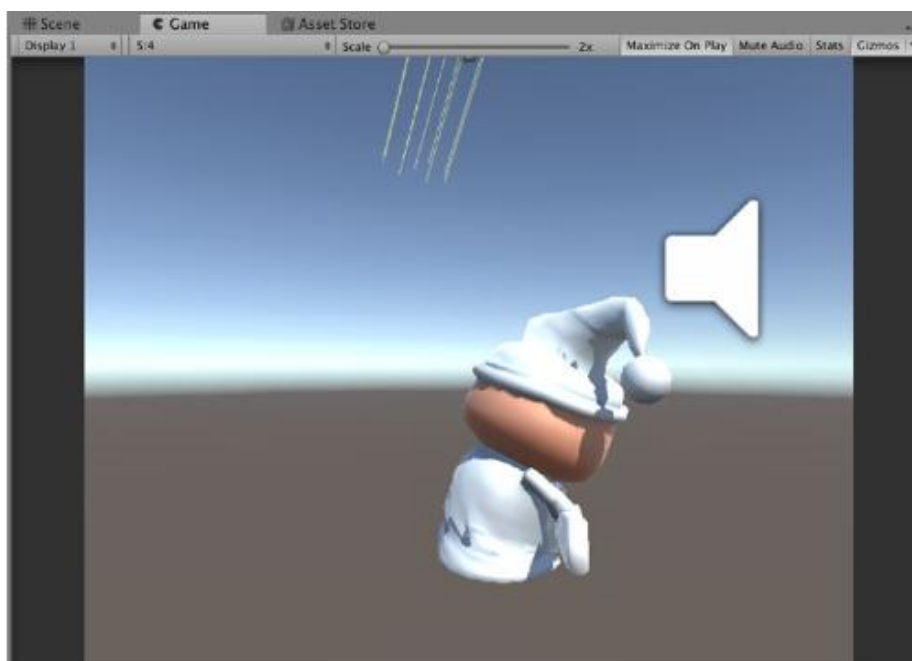
Tombol Statistik menampilkan statistik tentang adegan (Gambar 2.35) yang diperbarui saat permainan berjalan.



Gambar 2.35 Tampilan Permainan Dengan Statistik

Gizmos Tampilan Game

Tombol Gizmos mengaktifkan tampilan grafik diagnostik yang terkait dengan Komponen. Tampilan Game pada Gambar 2.36 memperlihatkan dua ikon yang merupakan gizmos untuk Sumber Audio. Daftar di sebelah kanan tombol Gizmos memungkinkan Anda memilih gizmos mana yang ingin ditampilkan.



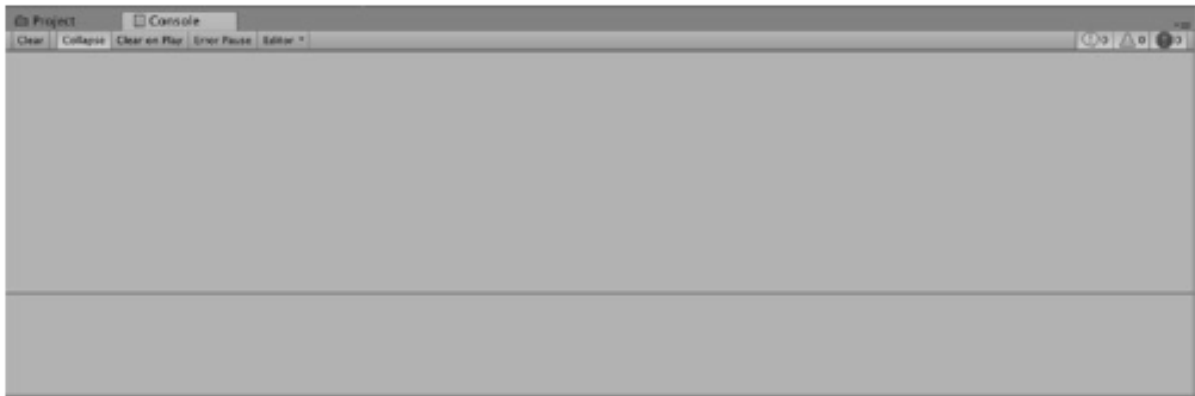
Gambar 2.36 Tampilan Game Dengan Gizmos

Baik Tampilan Game maupun Tampilan Adegan merupakan penggambaran adegan saat ini. Proyek Unity terdiri dari satu atau beberapa adegan, dan Editor Unity membuka satu adegan dalam satu waktu. Anggap proyek sebagai permainan dan adegan sebagai level (sebenarnya, beberapa fungsi skrip Unity yang beroperasi pada adegan menggunakan "level" dalam namanya).

GameObject Unity dibuat menarik dengan melampirkan Komponen, yang masing-masing menyediakan beberapa informasi atau perilaku tertentu. Di situlah Tampilan Inspektur berperan. Jika Anda memilih objek permainan dalam Tampilan Hierarki atau Tampilan Adegan, Tampilan Inspektur akan menampilkan komponen yang dilampirkan.

Tampilan Konsol

Tampilan yang tersisa di semua tata letak prasetel, Tampilan Konsol, mudah diabaikan tetapi cukup berguna (Gambar 2.37).



Gambar 2.37 Tampilan Konsol

Pesan informasi, peringatan, dan kesalahan muncul di Tampilan Konsol. Kesalahan berwarna merah, peringatan berwarna kuning, dan pesan informasi berwarna putih. Memilih pesan dari daftar akan menampilkannya dengan lebih detail di area bawah. Selain itu, area satu baris di bagian bawah Unity Editor menampilkan pesan Konsol terbaru, sehingga Anda selalu dapat melihat bahwa pesan telah dicatat meskipun tampilan Konsol tidak terlihat. Tip Pesan peringatan mudah diabaikan, tetapi Anda dapat mengabaikannya dengan risiko Anda sendiri. Pesan tersebut ada karena suatu alasan dan biasanya menunjukkan sesuatu yang harus diselesaikan.

Dan jika Anda membiarkan peringatan menumpuk, sulit untuk menyadari ketika peringatan yang benar-benar penting muncul. Konsol dapat menjadi berantakan dengan cepat. Anda dapat mengelola kekacauan itu dengan tiga tombol paling kiri di bagian atas Tampilan Konsol. Tombol sakelar Bersihkan menghapus semua pesan. Tombol sakelar Ciutkan menggabungkan pesan-pesan yang serupa. Tombol sakelar Bersihkan saat Putar akan menghapus semua pesan setiap kali Editor memasuki mode Putar. Tombol Jeda Kesalahan akan menyebabkan Editor berhenti saat ada pesan kesalahan, khususnya saat skrip memanggil `Log.LogError`.

Saat beroperasi di Editor, pesan log berakhir di log Editor, sementara pesan yang dihasilkan dari executable buatan Unity diarahkan ke log Pemain. Memilih Buka Log Pemain atau Buka Log Editor dari menu tampilan (klik ikon kecil di kanan atas Tampilan Konsol) akan memunculkan log tersebut, baik dalam file teks atau di aplikasi Konsol (Gambar 2.38).



Gambar 2.38 Log Unity Di Aplikasi Konsol Mac

Jelajahi Lebih Jauh

Kita telah sampai di akhir tur Unity ini. Di Bab 3, Anda akan mulai mempelajari beberapa fitur ARKit. Ini adalah bab pertama yang benar-benar mulai menggunakan Unity. Anda belum mulai membangun scene Anda sendiri (yang akan dimulai di Bab 3), tetapi Anda sudah bisa terbiasa dengan Unity Editor. Ada banyak sumber daya resmi Unity yang membahas lebih lanjut topik yang akan saya bahas.

2.9 MANUAL UNITY

Seperti yang Anda lihat, ada banyak antarmuka pengguna Unity, dan kami belum membahas semuanya. Ini saat yang tepat untuk mulai membaca Manual Unity, baik dari dalam Unity Editor (layar Selamat Datang atau menu Bantuan) atau di situs web Unity (<http://unity3d.com/>) di bawah tab Pelajari di bagian "Dokumentasi". Versi web cukup berguna saat Anda ingin mencari sesuatu atau sekadar membaca tentang Unity tanpa menjalankan Unity Editor di dekatnya.

Sebagian besar yang dibahas dalam bab ini sesuai dengan topik di bagian Dasar-dasar Unity dari Manual Unity, khususnya bagian tentang "Mempelajari Antarmuka," "Menyesuaikan Ruang Kerja Anda," "Menerbitkan Build," dan "Tombol Pintasan Unity." Kami langsung masuk ke bagian Lanjutan dari Manual Unity dan membahas dukungan Unity untuk kontrol versi. Itu dibahas lebih mendalam di halaman Manual Unity tentang "Menggunakan Kontrol Versi Eksternal dengan Unity."

Tutorial

Selain bagian "Dokumentasi", tab Pelajari di situs web Unity juga menyertakan bagian

"Tutorial" yang menampilkan serangkaian video Editor Pemula yang lengkap. Sesuai namanya, video ini memperkenalkan Editor Unity, dan sebenarnya rangkaian video tersebut mencakup banyak hal yang dibahas dalam bab ini, termasuk deskripsi tampilan yang paling penting (Tampilan Game, Tampilan Adegan, Tampilan Hirarki, Tampilan Inspektur, dan Tampilan Proyek) dan bahkan proses penerbitan versi.

Kontrol Versi

Meskipun saya hanya membahas kontrol versi secara singkat, dalam konteks menjelaskan cara menghapus metafile, topik tersebut layak dibahas lebih lanjut, karena sistem kontrol versi (atau VCS) sangat penting untuk pengembangan perangkat lunak (yang akan Anda sadari saat pertama kali kehilangan proyek atau tidak dapat mengingat perubahan apa yang Anda buat yang merusak game Anda!). Jika Anda sudah memiliki VCS favorit, Anda mungkin ingin menggunakannya dengan Unity, dan jika Anda belum pernah menggunakannya, maka Anda mungkin ingin mempertimbangkannya jika hanya untuk menyimpan versi lama proyek Anda jika Anda perlu melakukan rollback, dengan kemampuan untuk memeriksa perbedaan antar versi.

Di antara sistem kontrol versi, Perforce adalah alat komersial populer yang digunakan di studio game, dan Subversion (svn) memiliki sejarah panjang sebagai opsi sumber terbuka. Saat ini, sistem kontrol versi terdistribusi seperti Git dan Mercurial sedang menjadi tren. Saya menggunakan Mercurial di Bitbucket (<http://bitbucket.com/>) untuk proyek internal saya dan memposting proyek publik di GitHub, termasuk proyek untuk buku ini. Mengatakan dukungan Unity VCS tidak bergantung pada produk adalah cara lain untuk mengatakan Unity tidak memiliki sistem kontrol versi tertentu yang terintegrasi ke dalam Unity Editor.

Metafile, dan file adegan YAML untuk pengguna Unity Pro, hanya menyediakan kompatibilitas yang lebih baik dengan sistem kontrol versi berorientasi teks yang umum digunakan untuk kode sumber. Anda masih harus menjalankan sendiri operasi VCS di luar Unity. Anda dapat mengetahui lebih lanjut tentang YAML di <http://yaml.org/>. Menurut saya lebih mudah menggunakan aplikasi GitHub untuk Mac yang disediakan di situs web GitHub dan SourceTree untuk BitBucket, yang juga tersedia di situs web tersebut.

BAB 3

UNITY ARKIT

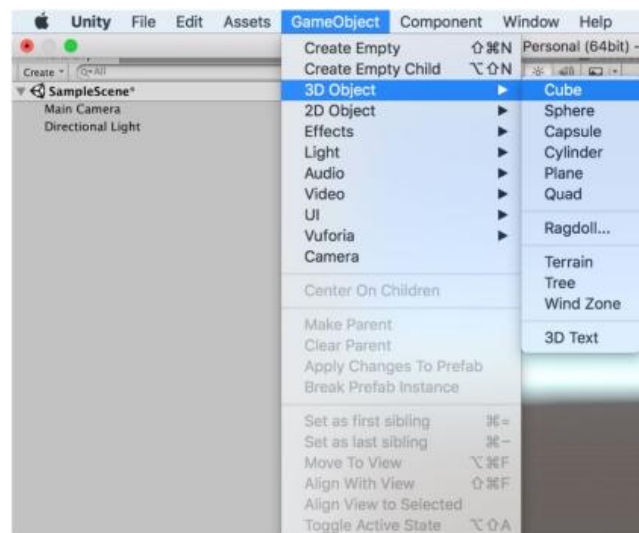
Sekarang setelah Anda menginstal Unity dan Plug-in Unity ARKit, saatnya untuk mempelajari lebih lanjut tentang alat ini dan membuat pengalaman AR pertama kita. Perlu dicatat, saya menyebutnya sebagai pengalaman dan bukan permainan, meskipun mungkin dapat digunakan sebagai permainan dalam beberapa hal; dari sudut pandang saya, permainan memerlukan beberapa elemen kunci, yang akan saya bahas secara terperinci di Bab 4.

Jika Anda belum membuka Proyek AR yang kita buat di bab terakhir, sekarang adalah waktu yang tepat untuk melakukannya. Jika Anda melihat di folder Proyek (Gambar 3.1) dan melihat di setiap folder, Anda akan melihat bahwa Plug-in Unity ARKit dilengkapi dengan sejumlah sumber daya yang akan kita jelajahi dan gunakan di seluruh buku ini.

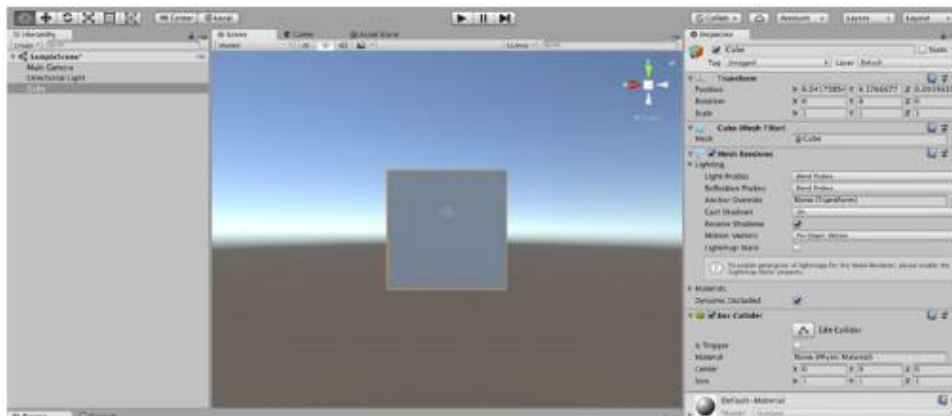
3.1 MEMBUAT ADEGAN

Tugas pertama yang akan kita lakukan adalah membuat GameObject dan meletakkannya di adegan kita dan melihatnya dalam AR. Dari Menu, pilih GameObject ► Objek 3D ► Kubus (Gambar 3.1).

Kita dapat memilih objek 3D mana pun, tetapi saya ingin Anda melihat Kubus dalam AR dan bergerak di sekitarnya dalam ruang 3D. Meskipun saya lebih suka Bola, Kubus 3D memiliki enam titik sudut (titik sudut poligon), dan ini akan lebih mudah dilihat. Namun, kami akan segera menambahkan objek permainan dengan bentuk yang berbeda. Setelah Anda menyelesaikan tugas ini, Anda akan melihat Cube yang menakjubkan di panel Scene (Gambar 3.2).

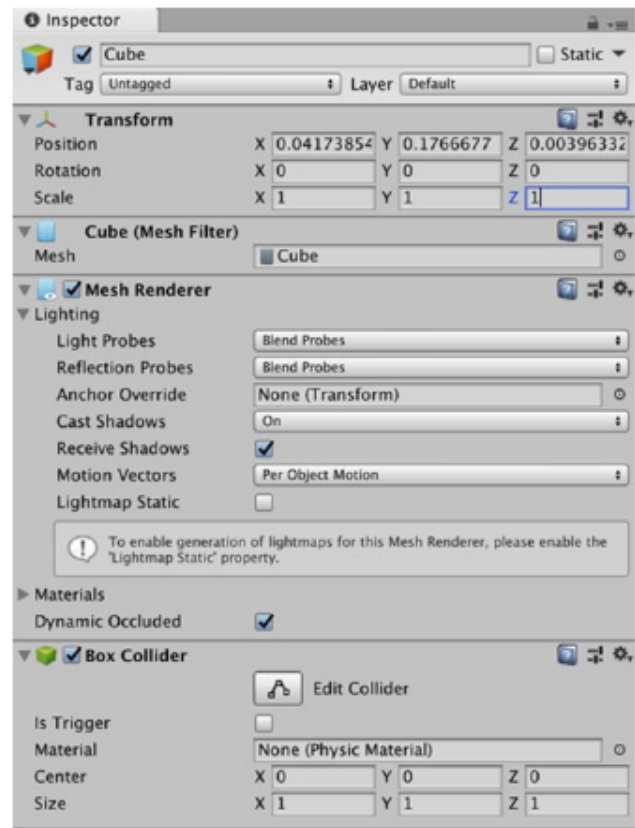


Gambar 3.1 Membuat GameObject 3D



Gambar 3.2 GameObject Cube Di Scene

Sekarang mari kita lihat di Inspector (Gambar 3.3). Jika posisi Transform Anda tidak sama dengan saya, jangan khawatir tentang ini untuk saat ini (kita akan tetap memindahkan GameObject). Perhatikan skalanya adalah 1,1,1. Unity menggunakan standar metrik untuk pengukuran, dan oleh karena itu 1 unit dalam ruang Unity setara dengan 1 meter dalam istilah dunia nyata. Sekarang untuk sebagian besar game tradisional (yang bukan AR), ini bukanlah sesuatu yang perlu dikhawatirkan oleh developer pemula (terlalu). Namun, karena kita akan memproyeksikan aset game ke dalam ruang dunia nyata, mendapatkan pengukuran dan skala yang benar menjadi sangat penting.



Gambar 3.3 Gameobject Cube Di Inspector

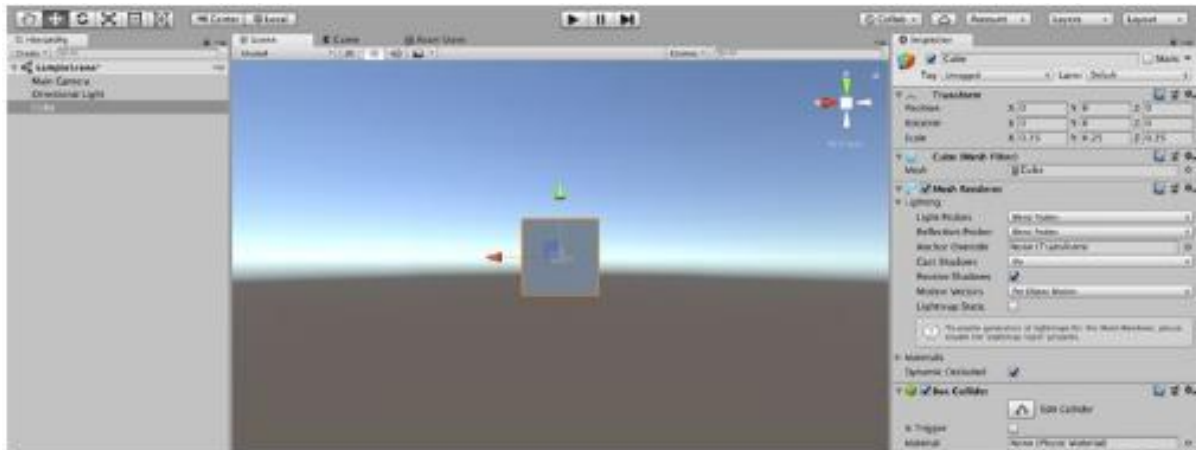


Gambar 3.4 Cube Gameobject Memperkecil Skala

Jadi, jika kita membiarkan skala Cube kita selebar 1 meter, tinggi 1 meter, dan dalam 1 meter, ini akan menjadi kubus yang cukup besar. Konvensi untuk menggambarkan koordinat dalam grafik 3D adalah X,Y,Z. Saya akan menggunakan standar ini di seluruh buku ini. Saya akan memperbesar Cube yang menakjubkan ini, sehingga bisa muat di apartemen saya yang sangat kecil. Mungkin beberapa pembaca akan terkejut bahwa sebagian besar penulis buku instruksional tidak tinggal di rumah mewah seharga jutaan dolar (setidaknya penulis ini tidak). Di Inspector, atur skala ke 0,25,0,25,0,25 (Gambar 3.4).

Sekarang jika Anda melihat Cube di Scene, ia akan (atau seharusnya) terlihat lebih kecil (Gambar 3.5). Jika Anda ingin melihat GameObject lebih dekat, Anda dapat menggerakkan kamera di tampilan Scene lebih dekat ke Cube. Ada dua cara utama untuk menggerakkan kamera tampilan Scene. Cara termudah adalah dengan memilih

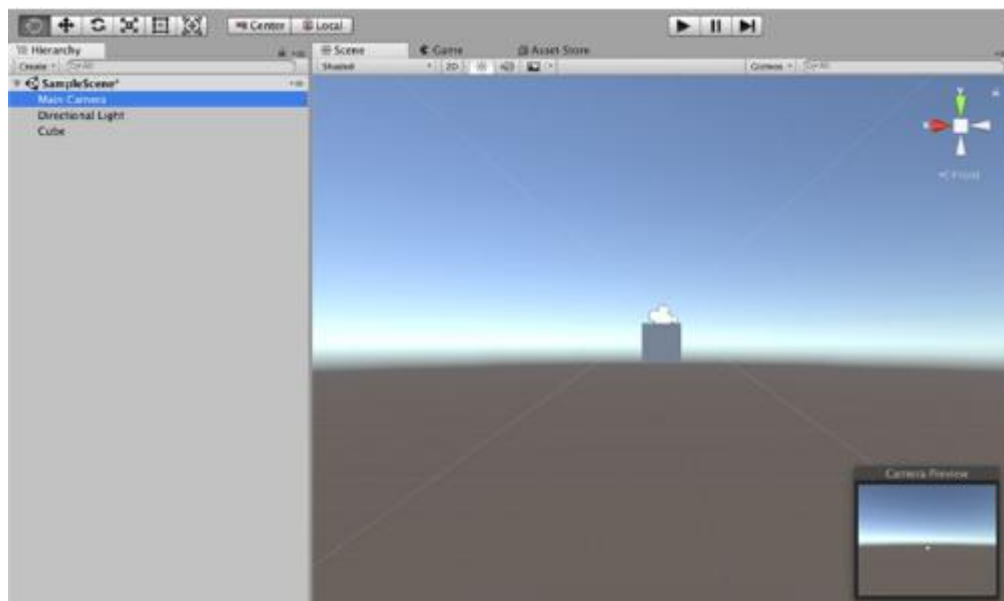
GameObject di panel Hierarchy dan menekan tombol f (saya ingat ini sebagai f untuk fokus). Cara lainnya adalah dengan menggunakan roda tengah mouse (jika Anda memiliki mouse 3 tombol) untuk menggerakkan kamera ke dalam atau ke luar atau mengetuk dua kali mouse (jika Anda memiliki Magic Mouse). Saya (sangat) kuno dan lebih suka menggunakan mouse 5 tombol.



Gambar 3.5 Tampilan Scene Dari Cube GameObject Pada Skala Yang Diperkecil

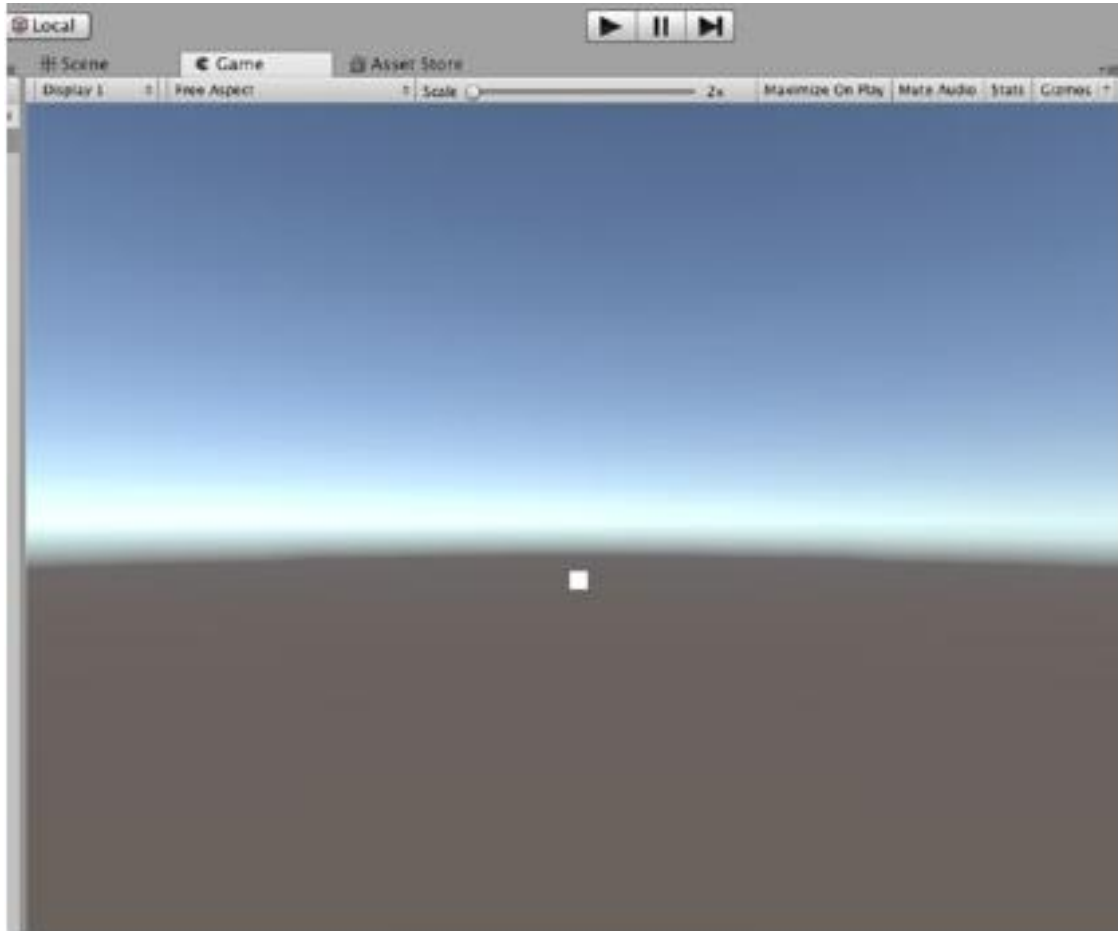
3.2 KAMERA

Sekarang saatnya untuk berbicara tentang kamera. Kita baru saja memindahkan kamera di tampilan Scene. Kamera ini memungkinkan Anda, pengembang, untuk melihat apa yang ada di Scene. Namun, jika Anda melihat di panel Hierarchy, Anda akan melihat GameObject Main Camera. Ini adalah kamera yang akan dilihat pemain. Anda akan melihat pada Gambar 3.5, ada ikon kamera di tampilan Scene saya. Di sinilah kamera Pemain diposisikan di scene.



Gambar 3.6 Pratinjau Kamera

Saya ingin Anda memilih Kamera Utama di panel Hierarki. Anda akan melihat jendela kecil di kanan bawah tampilan Adegan (Gambar 3.6). Jendela ini menampilkan tampilan dari kamera Pemain (Pratinjau Kamera). Meskipun jendela Pratinjau Kamera memberi kita gambaran yang baik tentang seperti apa tampilan adegan bagi pemain, untuk mendapatkan tampilan yang lebih baik, kita dapat memilih tab Tampilan permainan (Gambar 3.7).

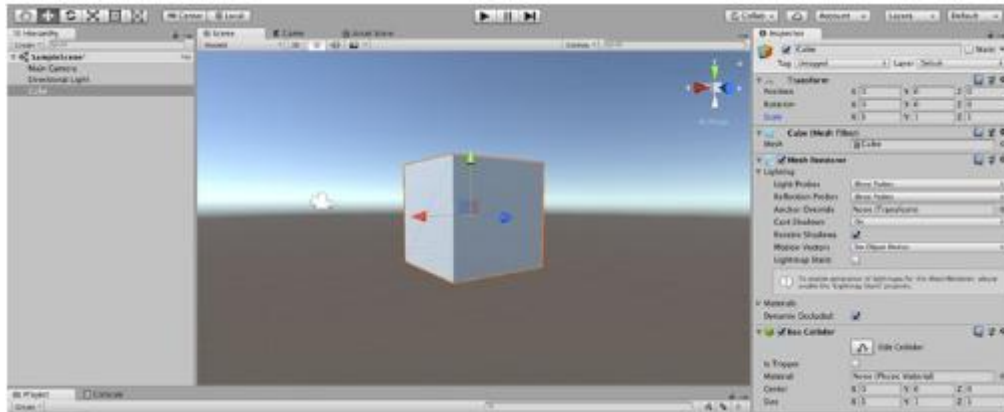


Gambar 3.7 Tampilan Game

Seperti yang kita lihat di bab sebelumnya, ada sejumlah pengaturan yang dapat kita gunakan (rasio aspek, skala, dan sebagainya). Kita dapat melihat di sini bahwa Cube kita yang menakjubkan agak jauh. Pertama-tama kita akan memindahkan Game Object, lalu kita akan memindahkan Main Camera.

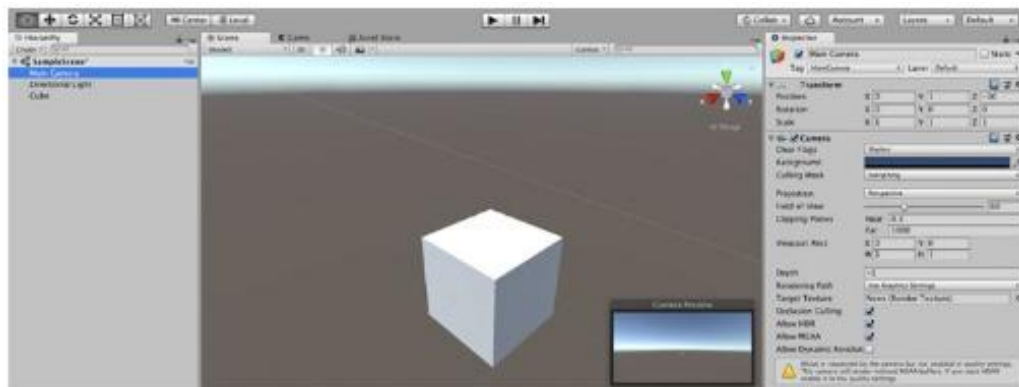
3.3 TRANSFORMASI

Ada beberapa cara untuk memindahkan (mengubah) GameObject di Unity. Cara pertama yang akan kita gunakan (dan merupakan metode utama yang saya gunakan) adalah dengan mengatur pengaturan transformasi di Inspector. Pertama, pilih GameObject yang ingin kita ubah. Kemudian, di inspector, atur posisi transformasi ke 0,0,0 (Gambar 3.8).



Gambar 3.8 GameObject Dengan Posisi Transform Yang Diperbarui

Sekarang mari kita lihat di Kamera Utama untuk memeriksa seperti apa tampilannya bagi pemain (Gambar 3.9).



Gambar 3.9 Pratinjau Kamera GameObject Dengan Posisi Transform Yang Disetel Ulang

Anda akan melihat bahwa GameObject masih agak terlalu jauh dari Kamera Utama. Di kiri atas layar (tepat di atas panel Hierarchy), terdapat enam ikon (Gambar 3.10).

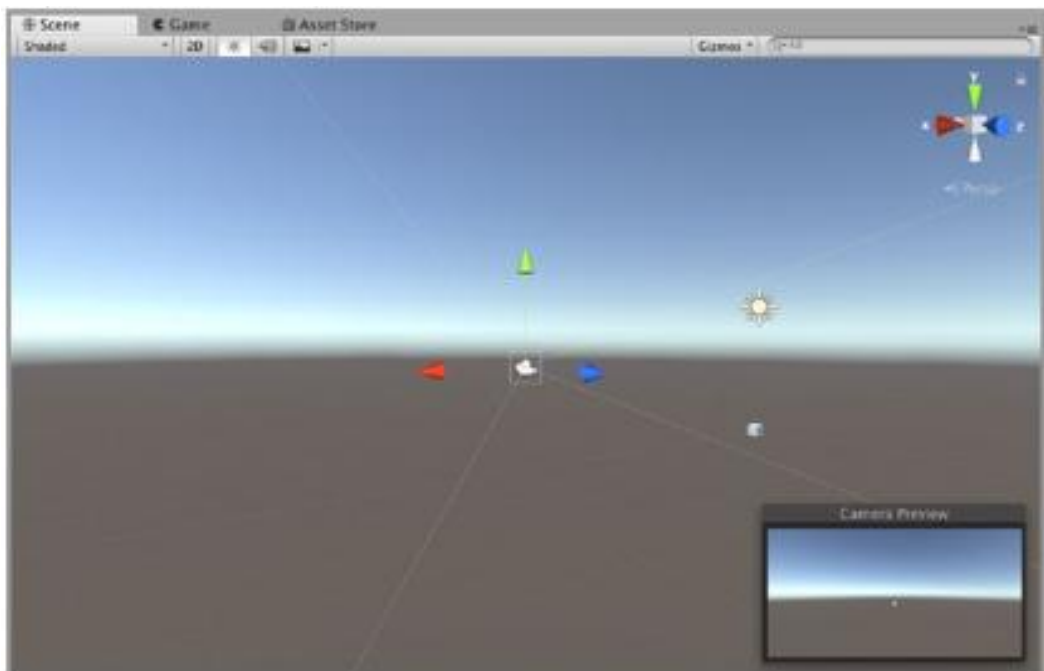


Gambar 3.10 Enam Ikon Transformasi

Ikon pertama adalah alat Tangan (pintasan Q), dan digunakan untuk mengubah (atau menggeser) kamera dalam tampilan adegan. Dengan alat ini dipilih, pilih GameObject mana pun dalam Hierarchy, gerakkan tetikus, dan perhatikan bahwa pengaturan transformasi GameObject tidak berubah, tetapi tampilan GameObject berubah. Sambil menahan tombol alt (opsi), alat Tangan memungkinkan Anda untuk mengorbitkan kamera di sekitar titik porosnya. Dengan menahan tombol kontrol, Anda dapat menggerakkan (atau menggeser) kamera lebih dekat atau lebih jauh dari GameObject.

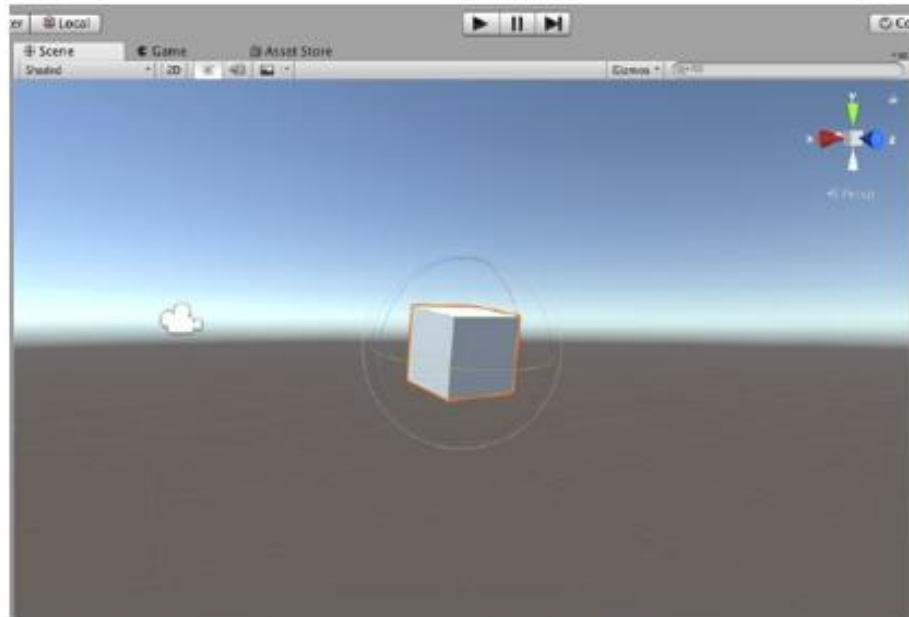
Ikon berikutnya adalah alat Pindah (pintasan W). Alat pindah, seperti yang mungkin dapat Anda duga, adalah untuk memindahkan posisi GameObject dalam adegan. Dengan

Kamera Utama dipilih, pilih alat Pindahkan (dengan memilih ikon Pindahkan atau menekan tombol W), dan Anda akan melihat bahwa dalam tampilan Adegan ikon Kamera Utama sekarang memiliki Gizmo bergerak (garis Merah, Hijau Biru dengan titik panah – Gambar 3.11). Pertama, pilih garis Hijau (sumbu Y) dan gerakkan Kamera Utama ke bawah. Perhatikan bahwa saat Anda memilih garis ini, garis panah lainnya berwarna abu-abu dan garis yang Anda pilih berubah menjadi kuning. Sekarang coba gerakkan kamera ke 0 pada sumbu Y. Anda akan melihat bahwa ini mungkin memerlukan waktu. Anda mungkin ingin memasukkan nol di Y di Inspektur jika Anda tidak dapat menempatkannya dengan benar (itulah sebabnya saya lebih suka mengetik nilai di Inspektur).



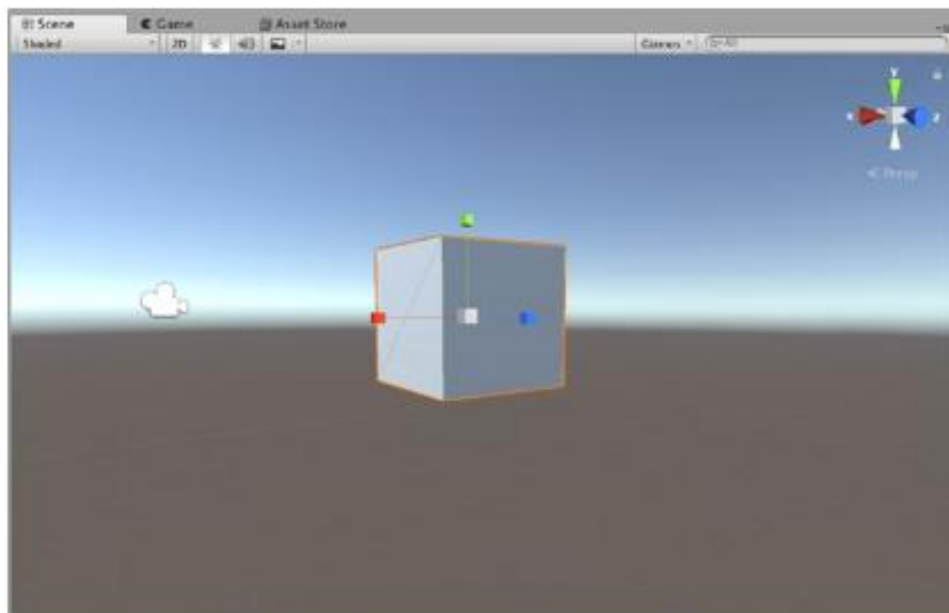
Gambar 3.11 Gizmo Untuk Bergerak

Alat berikutnya adalah alat Putar (pintasan E). Seperti yang mungkin sudah Anda duga, alat ini digunakan untuk memutar GameObject. Coba pilih Kubus lalu tekan tombol R. Anda akan melihat Gizmo Putar (Gambar 3.12). Dengan Gizmo Putar, Anda dapat mengubah rotasi GameObject dengan mengklik dan menyeret sumbu rangka kawat Gizmo Putar yang muncul di sekitarnya. Dengan Gizmo Putar, lingkaran merah, hijau, dan biru melakukan rotasi di sekitar sumbu merah, hijau, dan biru (merah adalah sumbu x, hijau adalah sumbu y, dan biru adalah sumbu z). Lingkaran luar digunakan untuk memutar GameObject di sekitar sumbu z tampilan Adegan.



Gambar 3.12 Gizmo Putar

Alat berikutnya adalah Scale toll (pintasan R). Alat skala digunakan untuk menskalakan atau mengubah skala GameObjects pada semua sumbu dengan memilih bagian tengah Scale Gizmo lalu menyeret tetikus (Gambar 3.13). Anda juga dapat menggunakan alat ini untuk menskalakan pada sumbu individual dengan memilih salah satu sumbu individual.



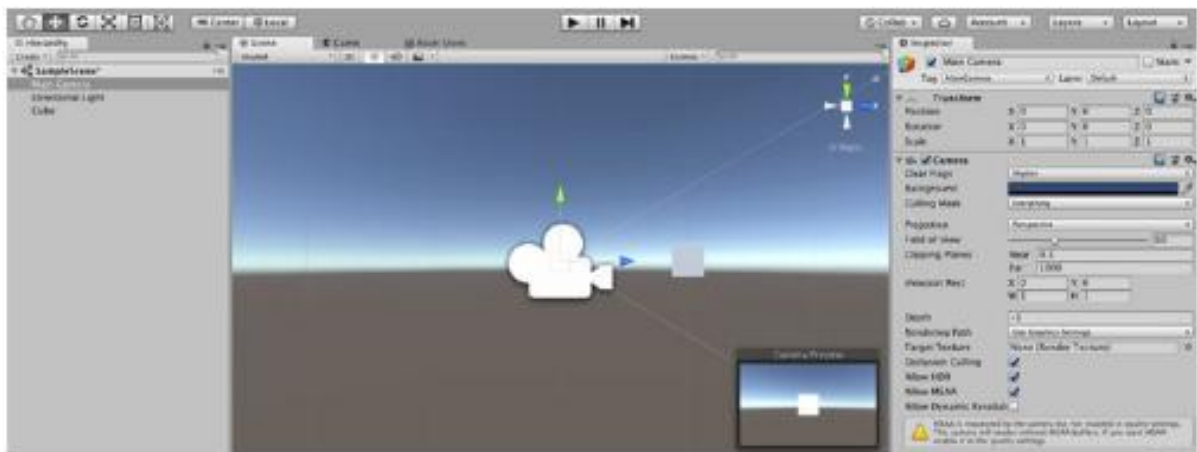
Gambar 3.13 Scale Gizmo

Penskalaan Kubus dengan GameObject Kubus yang dipilih akan menyetel posisi kubus, atau kubus telah dipindahkan saat kita menggunakan alat transformasi; sekarang adalah waktu yang tepat untuk menyetel atau mengatur ulang posisi ke 0,0,0 (Gambar 3.14).



Gambar 3.14 Pengaturan Posisi Gameobject Kubus

Sekarang kita perlu memposisikan Kamera Utama di titik asal (0,0,0). Jika Anda melihat posisi kubus saya, ini berarti kamera kita akan berada tepat di tengah Kubus kita. Jadi, pertama-tama mari kita pindahkan kubus tersebut. Atur pengaturan posisi transformasi Kubus ke 0,0,1 lalu pindahkan Kamera Utama ke 0,0,0 (Gambar 3.15). Sekarang kubus akan terlihat seperti diposisikan 1 meter dari kamera kita.



Gambar 3.15 Pengaturan Kamera Utama Yang Baru

3.4 PENGUJIAN

Sekarang saatnya untuk melihat bagaimana kubus menakjubkan kita akan terlihat di dunia nyata. Saat mengembangkan untuk iOS, kita perlu menggunakan Xcode sebelum kita dapat menguji atau menyebarkan game. Untuk menerbitkan game AR di iTunes, kita perlu mengirimkan game ke Apple untuk disetujui, dan kemudian setelah disetujui, kita dapat mengunduhnya dari toko iTunes dan melihat seperti apa tampilannya di perangkat kita. Dengan Xcode, kita dapat melihat pratinjau seperti apa tampilan game kita di perangkat kita, tetapi ini masih mengharuskan kita untuk membangun dan menjalankan game dan menggunakan Xcode untuk melihat pratinjau game.

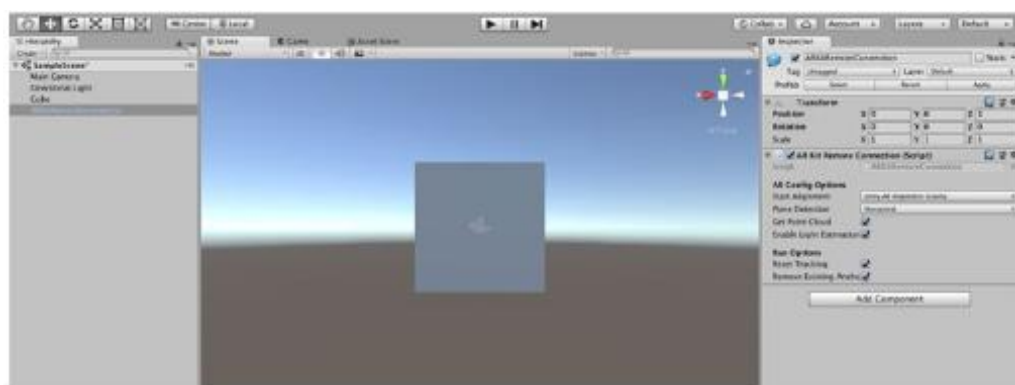
Seperti yang Anda bayangkan, ini adalah pengalaman yang sangat memakan waktu (dan sedikit membuat frustrasi). Unity datang untuk menyelamatkan dan memberi kita Unity Remote. Unity Remote adalah aplikasi (atau aplikasi) yang tersedia untuk perangkat iOS di toko aplikasi iTunes. Alat ini membantu kita menguji game di perangkat iOS tanpa perlu mengirimkan game ke toko iTunes. Namun, pada saat penulisan, Unity Remote (versi 5) tidak mendukung AR. Orang-orang hebat di Unity telah memikirkan tantangan ini dan telah menyertakan dalam Unity ARKit sebuah program kecil yang disebut UnityARKitRemote. Unity ARKitRemote menyediakan kita dengan alat yang dibutuhkan untuk menguji proyek AR kita pada perangkat iOS.

ARKit Remote

Di folder Project, ketik teks pencarian untuk menemukan ARKit Remote Prefab (Gambar 3.16). Sekarang seret berkas ini ke tab Hierarchy (Gambar 3.17).



Gambar 3.16 Mencari Arkit Remote



Gambar 3.17 Prefab Arkitremote Dalam Hierarki

Menyiapkan Kamera Utama

Sekarang di pengaturan Kamera Utama, atur tanda centang bening ke Kedalaman saja (Gambar 3.18).

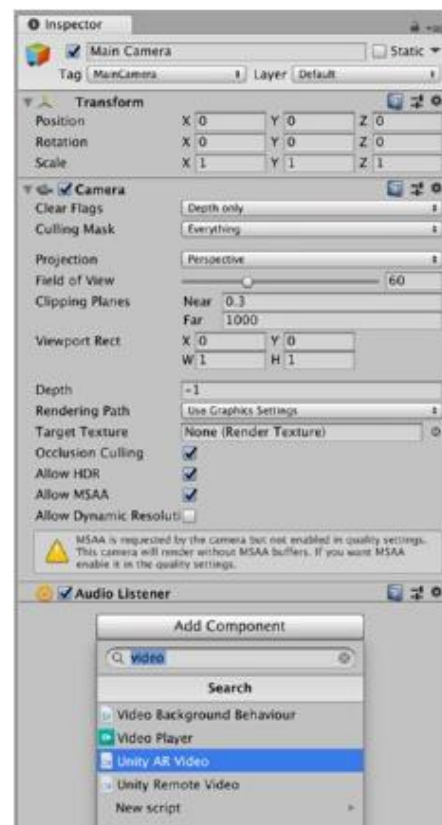


Gambar 3.18 Atur Clear Flag Kamera Ke Depth Only

Menambahkan Komponen

Sekarang kita akan menambahkan Komponen ke kamera kita. Dalam contoh ini, kita akan menambahkan skrip Unity AR Video. Untuk menambahkan skrip ini, dengan Kamera Utama dipilih, di Inspektur pilih Tambahkan Komponen. Di Inspektur, Anda akan melihat daftar komponen yang mungkin dapat Anda tambahkan (Gambar 3.18). Mungkin butuh waktu lama untuk menemukannya, jadi saya sarankan menggunakan bilah pencarian di menu Tambahkan Komponen untuk Skrip yang kita cari. Pada Gambar 3.19, di bilah pencarian, saya mencari video.

Setelah Anda menemukan skrip Unity AR Video, pilih skrip tersebut (klik tetikus sekali) dan sekarang Anda akan melihat komponen ini telah ditambahkan ke Kamera Utama (Gambar 3.20). Dalam skrip Unity AR Video, terdapat properti yang disebut Clear Material. Kita akan menambahkan material. Dalam properti Clear Materials skrip Unity



Gambar 3.19 Mencari Komponen Unity AR Video Script

AR Video, di sebelah kanan kotak properti, terdapat roda gigi kecil (Gambar 3.19); jika Anda memilih roda gigi ini, Anda akan melihat daftar semua material yang mungkin tersedia di folder proyek ini (Gambar 3-21). Sekali lagi, Anda dapat mencarinya secara manual, atau menggunakan bilah pencarian. Cari material YUV. Saya akan membahas lebih detail tentang apa yang dilakukan material ini dan mengapa kita menggunakannya di bab berikutnya.



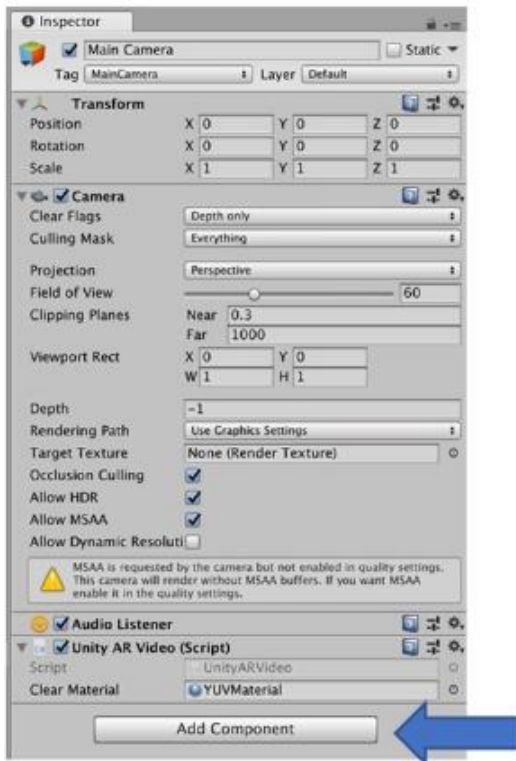
Gambar 3.20 Komponen Skrip Unity AR Video Telah Ditambahkan Ke Kamera Utama



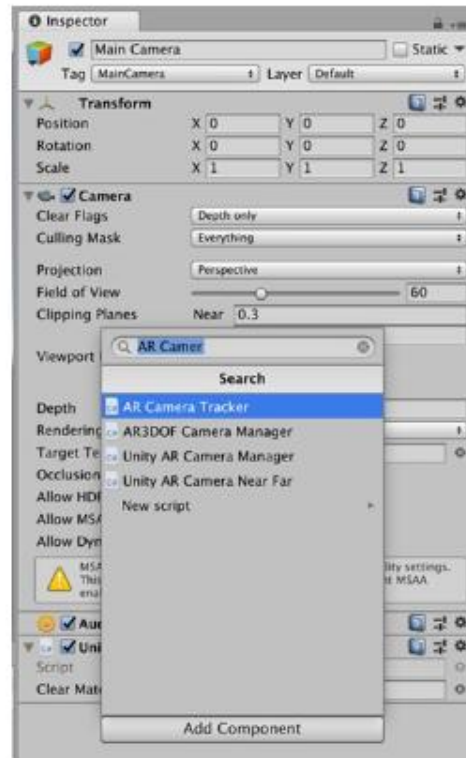
Gambar 3.21 Memilih Material YUV Untuk Material Bening

3.5 MELACAK PERGERAKAN PONSEL

Agar proyek AR terlihat nyata, kita perlu melacak pergerakan ponsel di dunia nyata dan memproyeksikan representasi akurat objek virtual di layar ponsel. Orang-orang hebat di Unity telah (sekali lagi) membuat hidup kita lebih mudah dan di Unity ARKit terdapat Skrip yang disebut Unity AR Camera Manager. Dengan mengikuti langkah-langkah yang kita lakukan untuk menambahkan skrip Unity AR Video, kita akan menambahkan Unity AR Camera Manager. Pertama-tama pilih Kamera Utama dan di Inspektur, pilih Tambahkan Komponen (Gambar 3.22); sekarang cari AR Camera Manager (Gambar 3.23), dan tambahkan ke Kamera Utama.



Gambar 3.22 Tambahkan Komponen



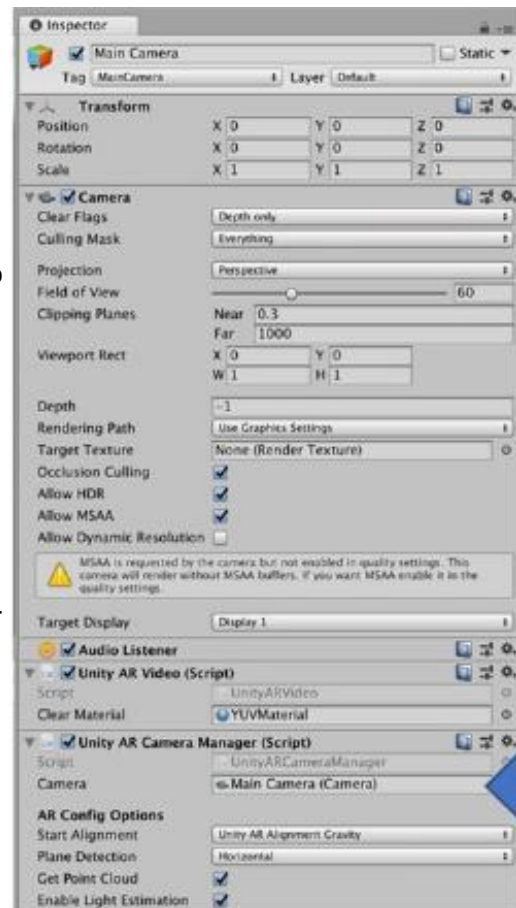
Gambar 3.23 Mencari Unity AR Camera Manager

Adalah praktik yang baik untuk menambahkan Kamera Utama ke Properti Kamera Terlacak dari Unity AR Camera Manager; ini akan memastikan bahwa AR Camera Manager menggunakan kamera yang benar. Namun, jika Anda tidak memilih ini, Unity AR Camera Manager akan memilihnya untuk Anda. Untuk menambahkan Kamera Utama ke Properti Kamera dari Unity AR Camera Manager, pilih Kamera Utama dari Hierarki dan seret ini ke properti Kamera Terlacak dari Unity AR Camera Manager (Gambar 3.24).

Bangun dan Jalankan

Sekarang kita siap untuk membangun dan menjalankan aplikasi kita. Pastikan Anda telah menghubungkan perangkat iOS ke Mac Anda, lalu pilih Bangun & Jalankan dari menu Berkas Unity (Gambar 3.25). Jika Anda belum

Gambar 3.24 Menetapkan Kamera Utama Sebagai Kamera Terlacak



mengunduh Xcode versi terbaru, Anda perlu melakukannya sekarang (ini akan memakan waktu cukup lama).

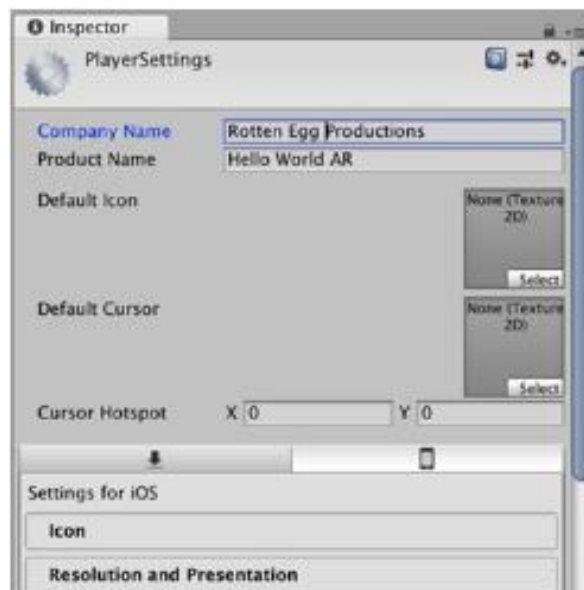


Gambar 3.25 Memilih Build & Run



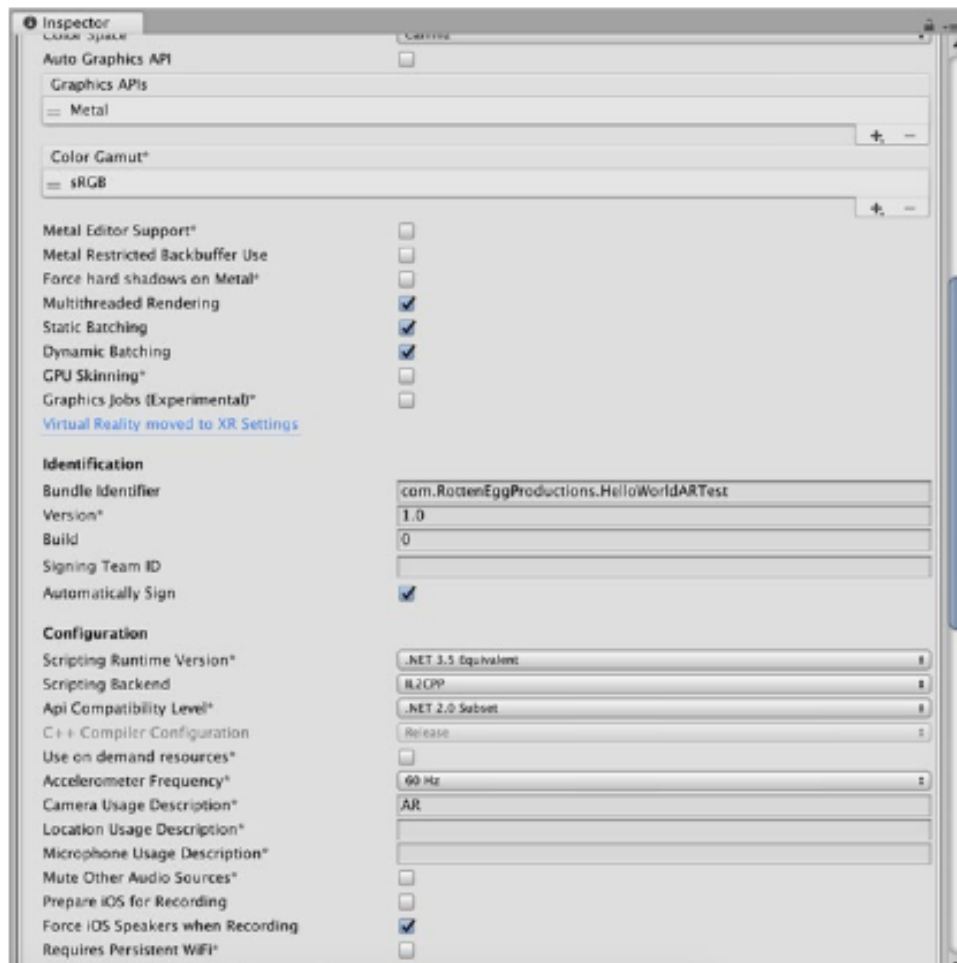
Gambar 3.26 Pengaturan Build

Di menu pengaturan Build, pertama-tama pilih Platform yang ingin Anda gunakan untuk membangunnya, yang dalam contoh kita adalah iOS. Selain itu, pilih kotak centang Development Build. Terakhir, sangat penting untuk hanya memilih adegan yang ingin Anda bangun. Jika adegan Anda saat ini tidak tercantum, klik tombol Add Open scenes (Gambar 3.26) dan hapus centang pada adegan lainnya. Dari layar Build Setting, pilih ikon Player Settings. Ini akan membuka Inspector for the Player Settings tempat kita akan memasukkan Company Name dan Product Name (Gambar 3.27).



Gambar 3.27 Tampilan Inspektur Playersettings

Gulir ke bawah menu untuk menemukan Bundle Identifier (Gambar 3.26). Penting untuk dicatat bahwa setelah Anda mendaftarkan pengenal bundel ke Personal Team di Xcode, pengenal bundel yang sama tidak dapat didaftarkan ke tim Apple Developer Program lain di masa mendatang. Ini berarti bahwa saat Anda menguji game menggunakan Apple ID gratis dan Personal Team, Anda harus memilih pengenal bundel yang hanya untuk pengujian - Anda tidak akan dapat menggunakan pengenal bundel yang sama untuk merilis game.

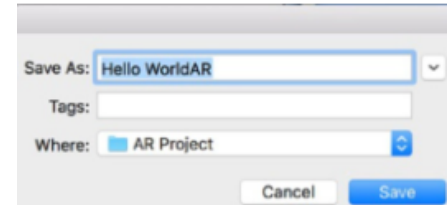


Gambar 3.28 Mengatur Bundle Identifier Di Playersettings

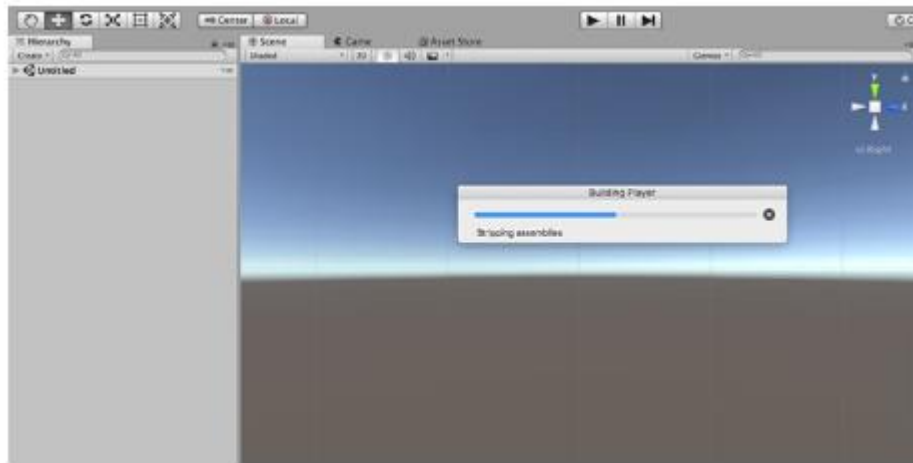
Solusi terbaik untuk melakukan ini adalah menambahkan "Test" di akhir pengenal bundel pengujian - misalnya, `com.yourCompanyName.yourAppNameTest`. Perlu dicatat juga bahwa pengenal bundel ditulis dalam gaya yang dikenal sebagai reverse-DNS. Karakter yang diterima adalah karakter alfanumerik, titik, dan tanda hubung. Dalam contoh saya (Gambar 3.28), saya telah menggunakan `com.RottenEggProductions.HelloWorldARTest` sebagai pengenal bundel (Anda akan memerlukan nama Anda sendiri). Jika Anda memiliki ID tim penandatanganan, Anda mungkin ingin menyertakannya juga. Namun, untuk tujuan pengujian, ini tidak diperlukan. Sekarang pilih ikon Build and Run.

Unity akan meminta Anda untuk menyimpan proyek. Tradisi saat mempelajari kursus Pemrograman Pengantar adalah memberi nama aplikasi pertama kita Hello World (jangan

tanya saya kenapa). Jadi, dalam tradisi ini, saya akan memberi nama Aplikasi AR pertama saya, Hello WorldAR. Perhatikan pada Gambar 3.29, saya menyimpan ini di folder yang sama dengan Proyek Unity saya. Beberapa orang mungkin berpendapat bahwa ini bukan praktik yang baik, tetapi cukup baik untuk saat ini.

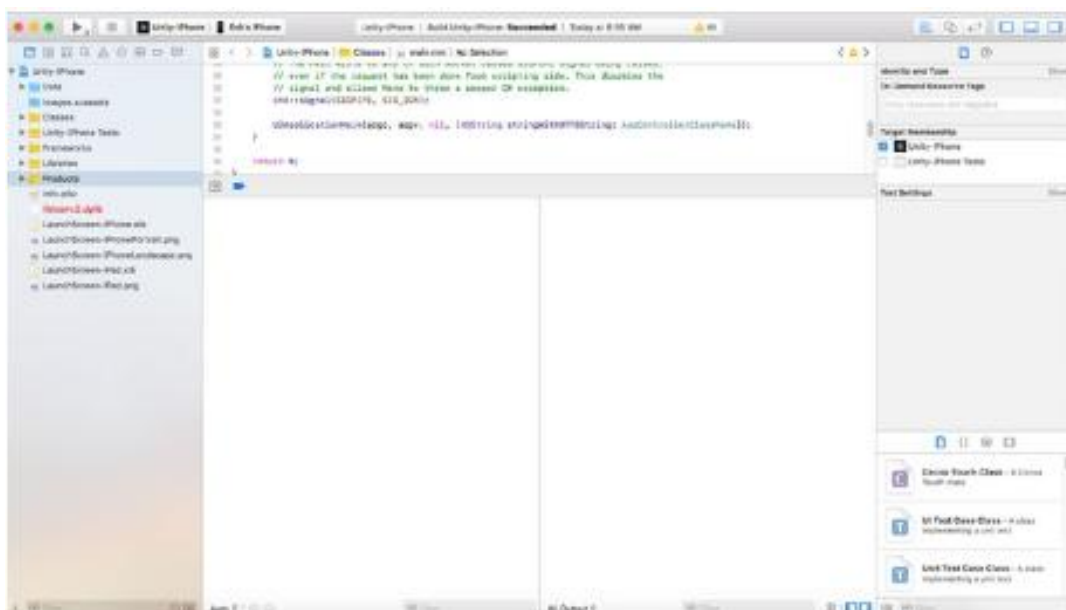


Gambar 3.29 Menyimpan Menu Hello Worldar



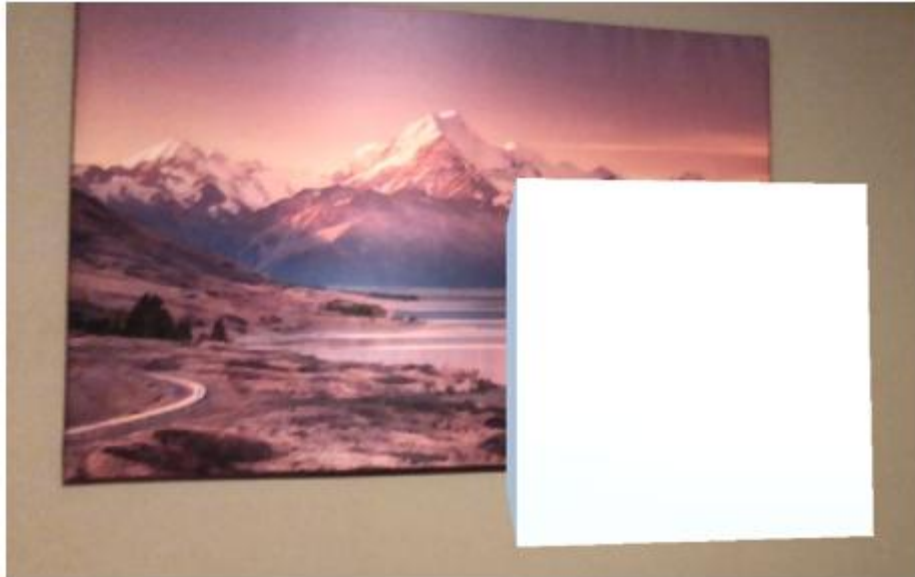
Gambar 3.30 Unity Sedang Mengkompilasi Aplikasi Kita

Setelah menyimpan file, Unity akan mulai mengompilasi aplikasi (Gambar 3.28) dan akhirnya akan membuka Xcode (Gambar 3.29). Saat Xcode terbuka (Gambar 3.31), pastikan Anda telah memilih perangkat yang benar (iPhone atau iPad), pilih tombol putar untuk meluncurkan game di perangkat Anda (Gambar 3.30). Jika Unity melaporkan kesalahan apa pun, pastikan untuk memeriksa apa kesalahannya dan atasi masalah ini sebelum mencoba lagi.



Gambar 3.31 Xcode

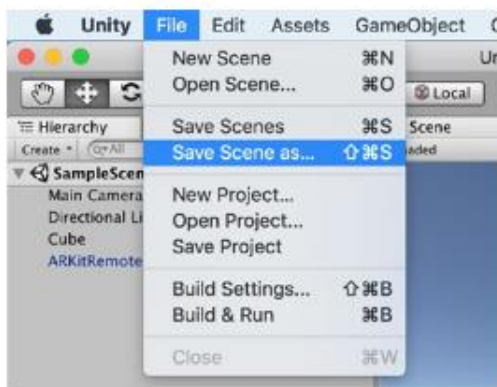
Anda akan diminta untuk mengizinkan Unity ARKit mengakses kamera Anda dan jika semuanya berfungsi, Anda akan melihat kubus menakjubkan Anda di dunia nyata. Dalam contoh saya, Anda dapat melihat kubus menakjubkan saya diproyeksikan di depan foto Gunung Aoraki (Gambar 3.32).



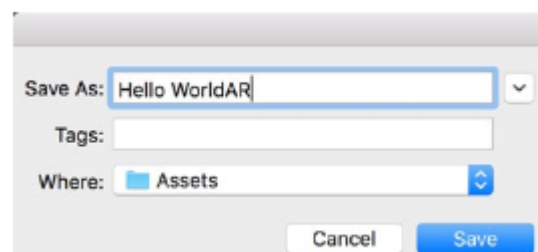
Gambar 3.32 Aplikasi Hello WorldAR Saya

Menyimpan Adegan

Sekarang saatnya untuk menyimpan adegan kita. Dari menu file, pilih File ► Save Scene As dan beri nama adegan ini (Gambar 3.33). Saya telah memilih Hello WorldAR sebagai nama Adegan ini (Gambar 3.34).



Gambar 3.33 Simpan Adegan Sebagai Menu



Gambar 3.34 Menyimpan Adegan Dan Memilih Lokasinya

Memahami Adegan

Sekarang mungkin saat yang tepat untuk membahas perbedaan antara adegan dan proyek Unity. Proyek Unity berisi semua adegan dan kode yang diperlukan yang mungkin

digunakan untuk permainan atau aplikasi. Adegan adalah elemen (atau komponen) dari proyek. Anggap proyek sebagai keseluruhan film dan adegan sebagai bagian dari film tersebut. Dalam permainan, adegan tersebut dapat berupa menu, level, kredit, dan sebagainya.

3.6 MEMPERKENALKAN VISUAL INERTIAL ODOMETRY

Sekarang kita akan melihat beberapa alat penting lainnya yang digunakan untuk membuat game AR. Dalam proyek Hello WorldAR, kita membuat kubus yang diposisikan di depan kamera iPhone dan tetap di sana saat kita memindahkan posisi kamera. Bagaimana kamera mengetahui di mana posisinya? Seperti yang mungkin sudah Anda ketahui, iPhone memiliki beberapa cara yang cukup keren untuk mengetahui di mana posisinya. Yang paling sering saya gunakan adalah accelerometer. Accelerometer memungkinkan iPhone mengetahui posisinya dalam 3 Sumbu (X, Y, Z).

Ini sangat berguna untuk beralih antara mode potret dan lanskap. Alat lain yang saya gunakan adalah kompas (atau magnetometer), dan seperti yang mungkin sudah Anda ketahui, ini sangat berguna untuk navigasi. Alat terakhir adalah giroskop. Giroskop melacak rotasi atau putaran iPhone. Meskipun ini adalah alat yang hebat untuk navigasi, alat ini tidak memiliki tingkat presisi yang diperlukan untuk melacak pergerakan ponsel dalam AR.

Untuk melacak pergerakan iPhone yang dibutuhkan untuk AR, Apple baru-baru ini menyertakan beberapa teknologi di kamera ponsel. Dengan menggabungkan informasi Visual (dari kamera) dan informasi Inersia (dari accelerometer dan giroskop), posisi iPhone dapat diukur secara akurat.

Poin Fitur

Jadi, bagaimana kamera melacak posisi ponsel? Pertanyaan bagus! Kamera di iPhone (saat ini iPhone 8 atau yang lebih tinggi) cukup pintar untuk mengidentifikasi titik-titik utama (atau titik fitur) di dunia nyata dan saat kamera digerakkan, melacak di mana titik-titik ini berada. Proses ini memerlukan matematika yang cukup mengesankan, tetapi iPhone yang lebih baru memiliki daya pemrosesan yang cukup untuk melakukan ini.

Point Clouds

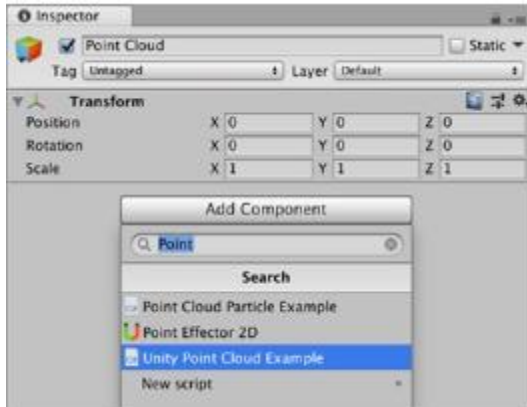
Unity ARKit menyertakan prefab untuk membantu ponsel mengidentifikasi titik fitur di dunia nyata. Yang pertama akan kita gunakan adalah PointCloud Prefab. Dengan aplikasi Hello WorldAR yang terbuka di Unity, kita akan membuat GameObject kosong. Dari menu file, pilih GameObject ► Create Empty (Gambar 3.35).

Dengan GameObject Kosong yang Dipilih di Inspektur, ubah namanya menjadi Point Cloud. Sekarang tambahkan komponen. Di bilah pencarian, cari Unity Point Cloud Example dan tambahkan ini ke GameObject Point Cloud (Gambar 3.36).

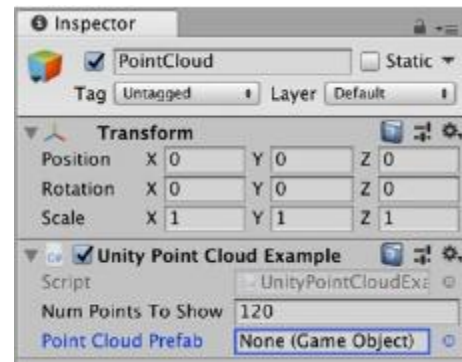


Gambar 3.35 Membuat GameObject Kosong

Dengan skrip Unity Point Cloud Example yang ditambahkan, sekarang atur jumlah Max Points to Show menjadi 120 (Gambar 3.37). Anda dapat mengatur Point Cloud sebanyak yang Anda suka.

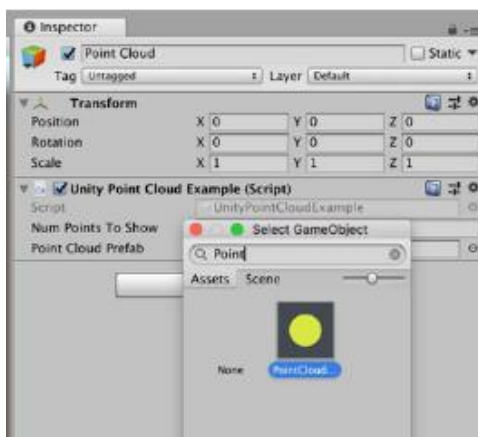


Gambar 3.36 Mencari Contoh Point Cloud

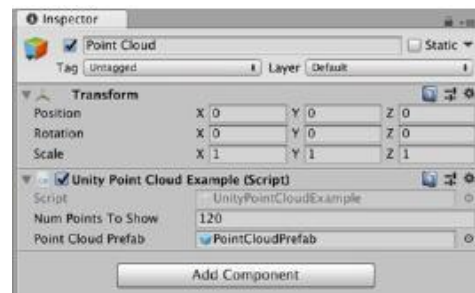


Gambar 3.37 Mengatur Max Points To Show

Sekarang kita perlu menambahkan Point Cloud Particle Prefab ke Point Cloud. Pilih roda gigi kecil di sebelah kanan kotak Point Cloud Prefab lalu cari Prefab (Gambar 3.38). Pilih dan seret PointCloudPrefab ke kotak Point Cloud Particle Prefab di skrip Point Cloud Particle Example (Gambar 3.39).



Gambar 3.38 Mencari Pointcloud Prefab



Gambar 3.39 Set Pointcloudprefab Sebagai Point Cloud Prefab

Pengujian

Sekarang kita akan menguji Point Cloud kita. Ketika kita menguji aplikasi Hello WorldAR kita, kita melalui tugas yang panjang (dan mungkin membosankan) untuk membangun Aplikasi, meluncurkannya di Xcode, dan kemudian akhirnya dapat melihat Aplikasi kita di perangkat iOS kita. Orang-orang hebat di Unity telah memikirkan hal ini dan telah menciptakan cara bagi kita untuk mengurangi waktu pengujian pengembangan kita.

Di Unity ARKit, ada Adegan yang akan memungkinkan kita untuk melihat pratinjau pembangunan di tab Game Unity. Jika Anda ingin melalui proses menggunakan Xcode setiap kali Anda ingin melihat pratinjau pengembangan, itu bagus. Namun, saya akan menunjukkan

kepada Anda cara yang lebih efisien yang mungkin Anda hargai.

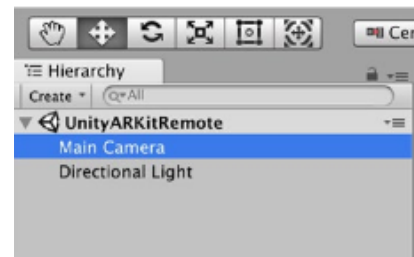
Unity ARKitRemote

Karena Unity Remote Connection saat ini tidak mendukung AR, kita perlu membangun dan menyebarkan Aplikasi ke perangkat iOS kita. Orang-orang hebat di Unity telah menyertakan dalam Unity ARKit, sebuah adegan yang disebut ARKit Remote. Anda dapat menemukannya di folder Project menggunakan bilah pencarian. Pada Gambar 3-40, saya telah menggunakan string pencarian remote.



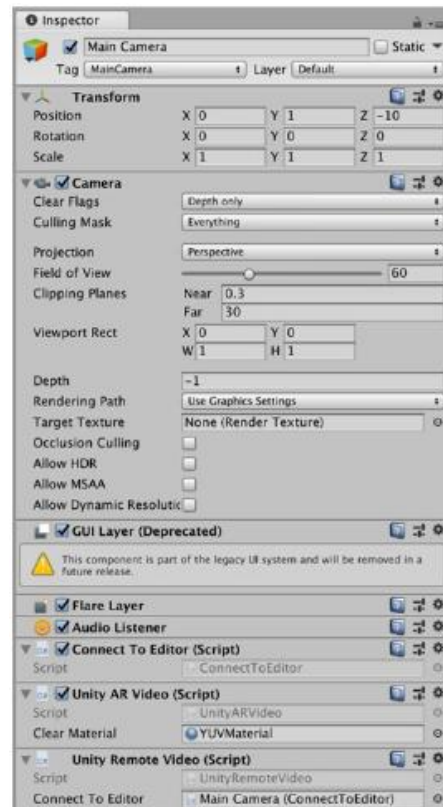
Gambar 3.40 Mencari Scene UnityARKitRemote

Klik dua kali Scene untuk membukanya. Jika Anda belum menyimpan perubahan apa pun pada scene saat ini, Anda akan diminta untuk menyimpannya sebelum Unity membuka scene lainnya. Anda akan melihat bahwa ini adalah Scene yang sangat sederhana yang terdiri dari Kamera Utama dan Lampu Terarah (Gambar 3.41).



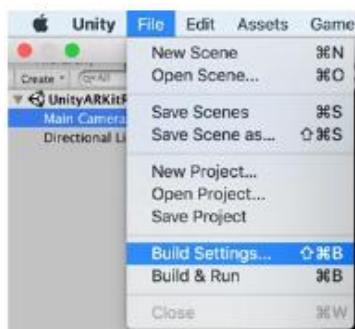
Gambar 3.41 UnityARKitRemote

Jika Anda memilih Kamera Utama dalam Hierarki, Anda akan melihat bahwa Kamera Utama memiliki beberapa skrip yang ditambahkan (Gambar 3.42). Skrip ini akan memungkinkan kamera ponsel melacak posisinya serta memungkinkan kita melihat tampilan kamera di Editor Unity.



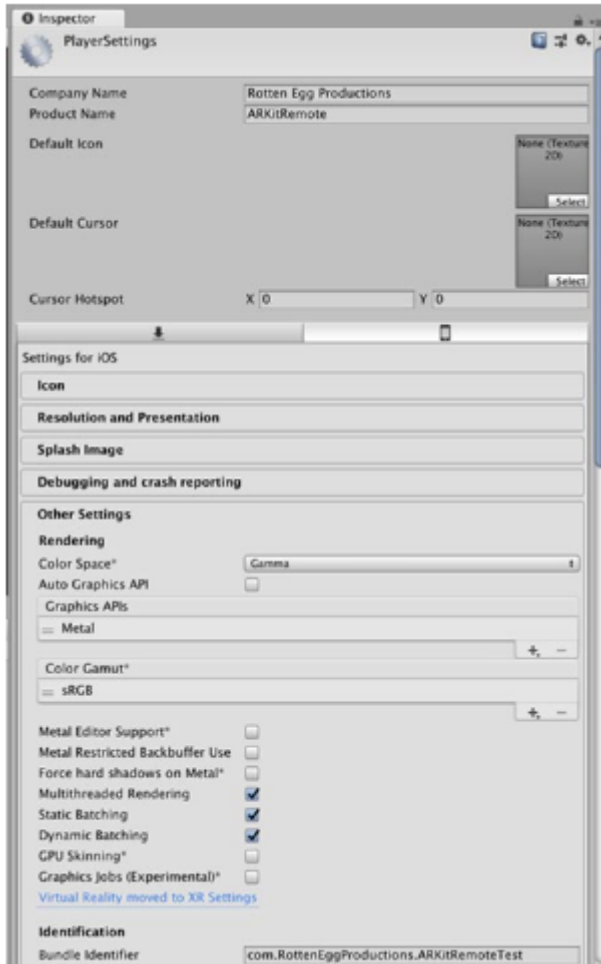
Gambar 3.42 Skrip Kamera Utama

Sama seperti yang kita lakukan dengan aplikasi Hello WorldAR yang kita buat, kita perlu menerapkan Aplikasi ini ke ponsel kita. Dari menu file, pilih File ► Build Settings (Gambar 3.43).

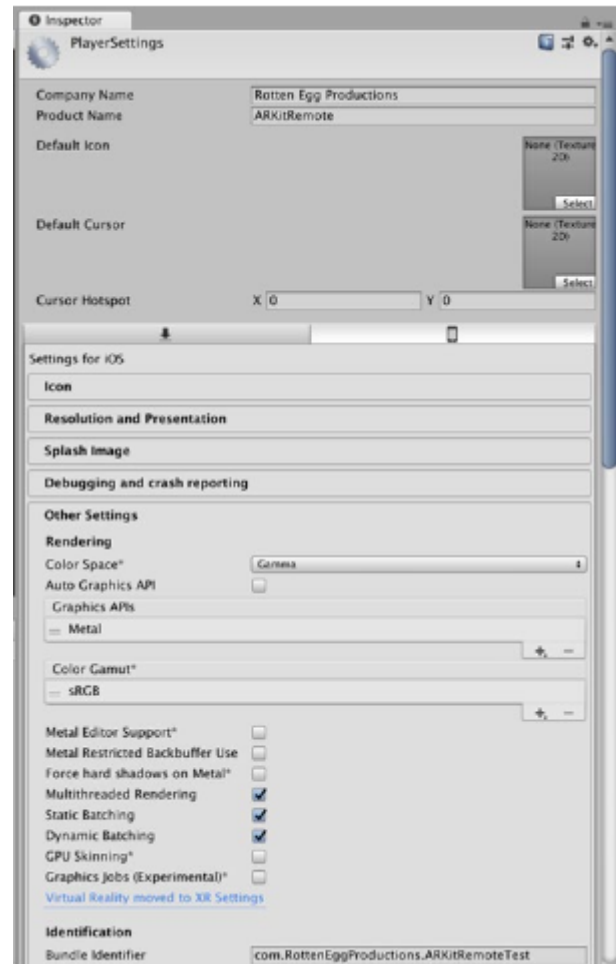


Gambar 3.43 Memilih Pengaturan Build

Di menu Pengaturan Build, pilih menu Pengaturan Pemain dan buat perubahan pada Nama Produk dan Pengenal Bundel (Gambar 3.44).



Gambar 3.44 Tampilan Inspektur
Playersettings



Gambar 3.45 Pengaturan Pengenal Bundel

Perlu dicatat, saya telah mengubah Nama Produk menjadi ARKitRemote dan Pengenal Bundel menjadi com.RottenEggProductions.ARKitRemoteTest (Gambar 3.45).

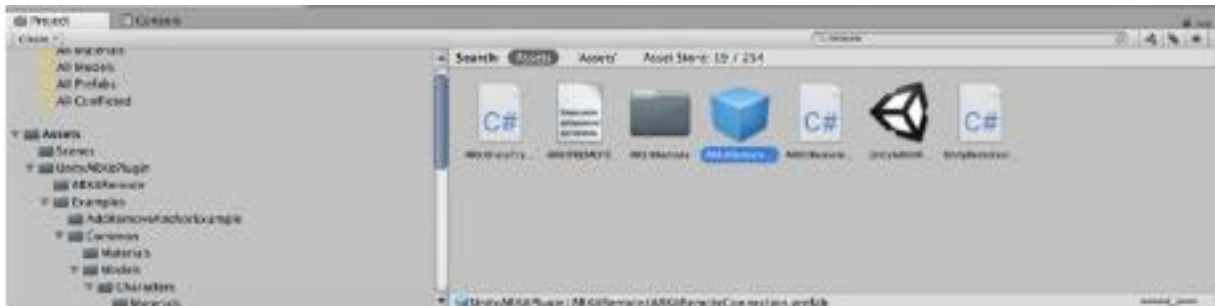


Gambar 3.46 Arkitremote Terpasang Di Iphone Saya

Pastikan Anda memiliki Scene yang tepat untuk membangun dan pilih Build and Run dari Build Menu. Sekarang Unity akan mengompilasi Aplikasi kita dan menjalankan Xcode dan Xcode akan menyebarkannya ke Perangkat kita (Gambar 3.46). Sekarang kita perlu menambahkan ARKit Connection Prefab ke proyek Hello WorldAR kita. Buka Scene Hello WorldAR Anda (klik dua kali).

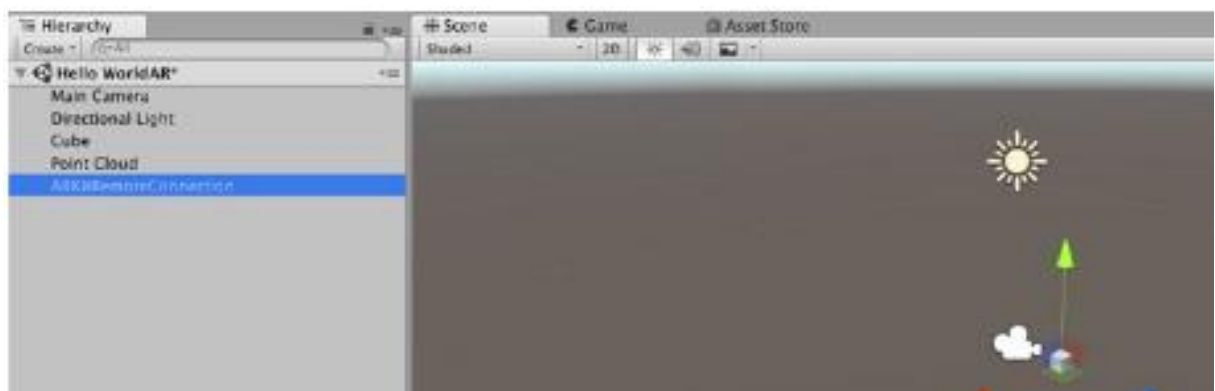
Menggunakan ARKit Remote Connection

Di Folder Proyek, cari Prefab bernama ARKitRemoteConnection (Gambar 3.47).



Gambar 3.47 Mencari Prefab Unityarkitremoteconnection

Pilih Prefab ini dan tambahkan ke Scene (seret ke Hierarchy) (Gambar 3.48).



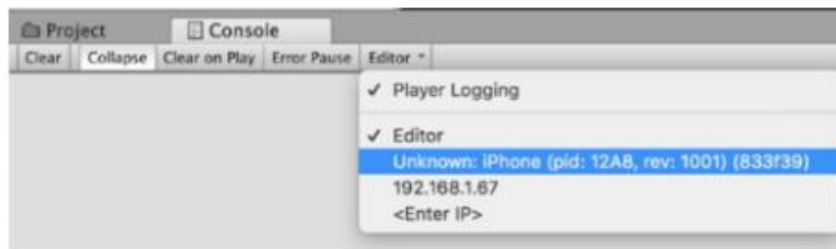
Gambar 3.48 Unityarkitremoteconnection Di Scene

Sekarang kita akan membuka ARKitRemote di perangkat iOS kita. Di Unity, buka scene Hello World AR lalu pilih tab Game dan tekan tombol Play. Unity akan meminta Anda untuk terhubung ke pemain di menu konsol (Gambar 3.49).



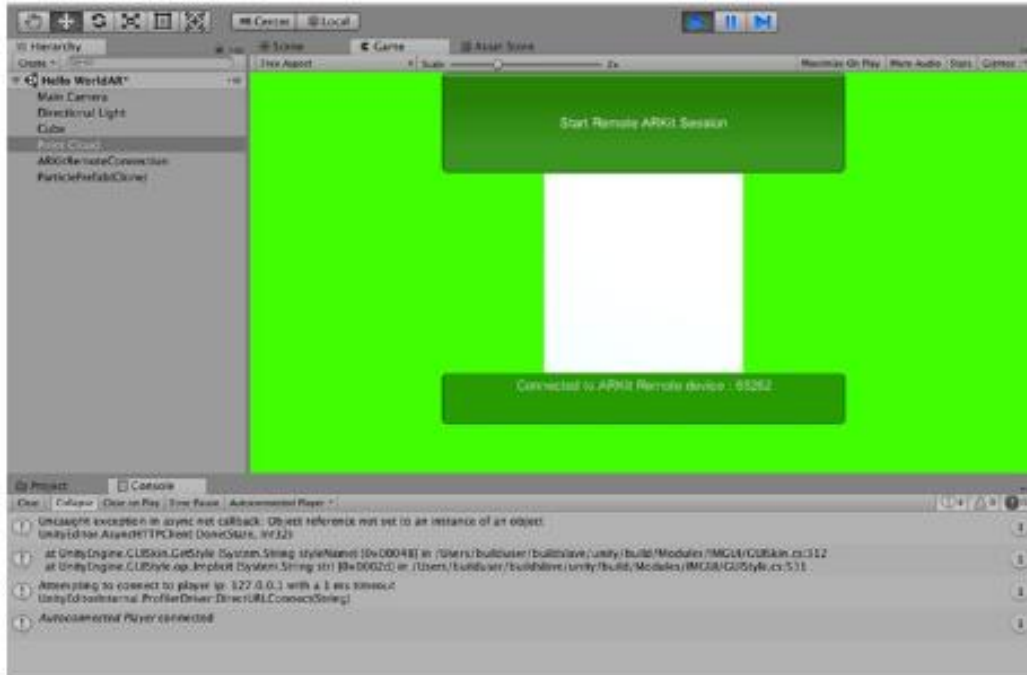
Gambar 3.49 Pesan Unity Connect To Player

Di tabel Console, pilih Editor dan pilih perangkat iOS Anda (Gambar 3.50).

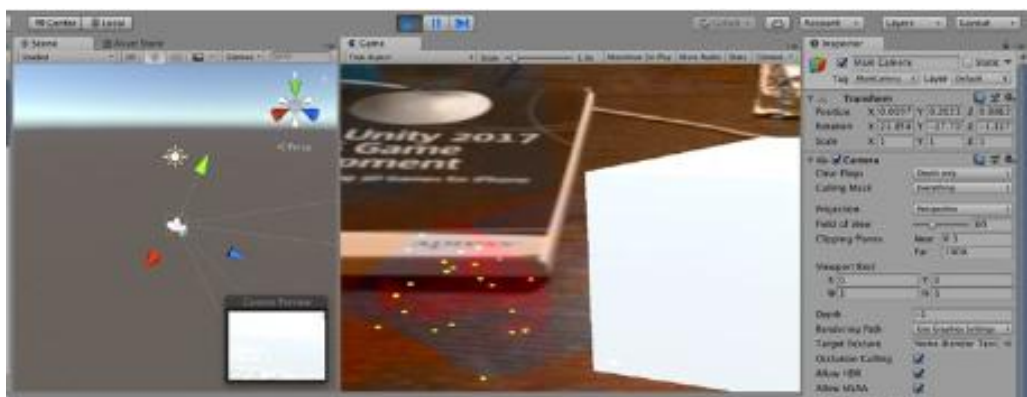


Gambar 3.50 Memilih Perangkat Ios Di Menu Console

Kemudian Unity akan meminta Anda untuk Start Remote ARKit Session (Gambar 3.51). Klik ikon di layar Game di Unity dan aplikasi Anda sekarang dapat berjalan di perangkat iOS Anda. Pada Gambar 352, Anda dapat melihat bahwa saya telah berhasil melihat aplikasi Hello WorldAR di iPhone saya dan melihat Scene di Unity. Jika Anda menggerakkan perangkat, Anda akan melihat Kamera bergerak di tab Unity Scene dan gambar berubah di tab Game. Saya ingin menunjukkan bahwa prosesnya akan sedikit lambat (lag), tetapi ARKitRemote saat ini merupakan cara terbaik untuk mengembangkan dan menguji pengembangan Unity AR untuk iOS.



Gambar 3.51 Perintah Mulai Sesi Arkit Jarak Jauh

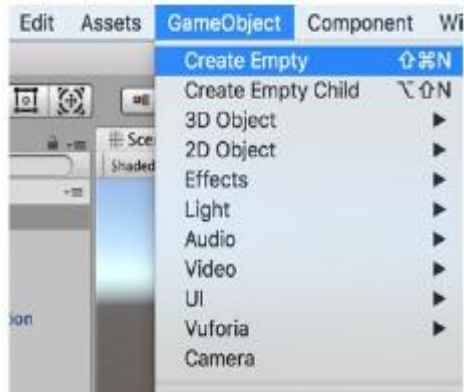


Gambar 3.52 Kubus Dan Point Cloud Dalam Tampilan Game

Meskipun sistem titik awan sangat bagus untuk mengidentifikasi objek yang tidak simetris (seperti sofa), sistem ini akan lebih efisien dan efektif jika kita menggunakan fitur lain yang disediakan dalam UnityARKit yang akan membantu mengidentifikasi bidang datar (seperti lantai, dinding, atau meja).

Visualisasi Bidang

Pertama, buat GameObject kosong lain yang akan berisi skrip yang dibutuhkan. Untuk membuat GameObject Kosong, dari menu, pilih GameObject ► Buat Kosong (Gambar 3.53). Setelah GameObject dipilih, pada Inspector, beri nama Created Planes (Gambar 3-54).

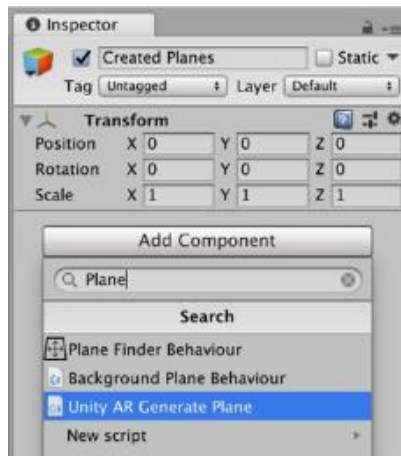


Gambar 3.53 Membuat Game Object Kosong

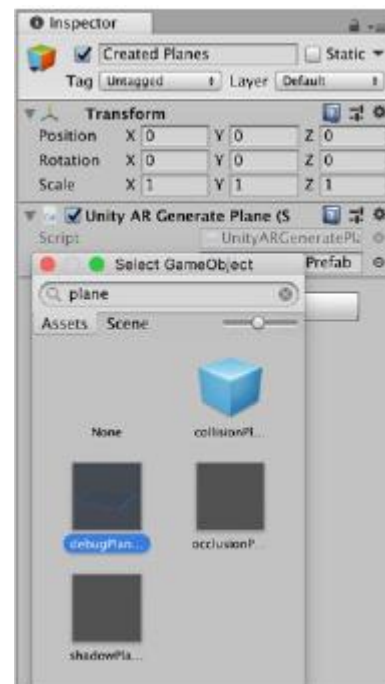


Gambar 3.54 Memberi Nama Gameobject Created Planes Yang Baru

Sekarang kita perlu menambahkan komponen. Pilih tombol Add Component di Created Planes Inspector dan di bilah pencarian, cari skrip Unity AR Generate Plane (Gambar 3.55).



Gambar 3.55 Mencari Skrip Unity AR Generate Plane



Gambar 3.56 Mencari debugPlanePrefab

Dengan skrip Unity AR Generated Plane yang disertakan dalam GameObject Created Planes, dalam pengaturan Plane Prefab dari skrip Unity AR Generated Plane, pilih roda gigi kecil di sebelah kanan kotak opsi dan cari serta pilih DebugPlanePrefab (Gambar 3.56).

Pengujian

Sekarang dengan perangkat iOS Anda terhubung ke Mac, jalankan Aplikasi UnityARKitRemote. Sekarang di Unity, hubungkan ke perangkat Anda di Editor dan klik tombol Play. Dengan perangkat iOS Anda, putar kamera untuk melihat permukaan datar, dan Anda akan melihat titik awan dan persegi panjang di layar tampilan Game Unity (Gambar 3.57).



Gambar 3.57 Point Cloud Dan Created Plane

Baik Point Cloud maupun Created Plane akan berguna dalam permainan yang kita buat untuk menemukan dan melacak titik referensi dari kamera iOS. Jika Anda merasa berani, Anda mungkin ingin mencoba menerapkan versi final ke perangkat iOS Anda. Namun, saat ini, Cube kita belum sepenuhnya siap untuk benar-benar menggunakan pelacakan titik referensi kamera.

Ringkasan

Dalam bab ini, kita telah mempelajari tentang penggunaan Unity ARKit dan mengonfigurasi perangkat iOS kita untuk melihat pratinjau aplikasi dan permainan kita secara real time. Kita juga mempelajari beberapa teori tentang bagaimana Unity ARKit dapat melacak posisi perangkat. Terakhir, kita menggunakan beberapa alat untuk membantu kita melacak gerakan halus perangkat secara akurat.

BAB 4

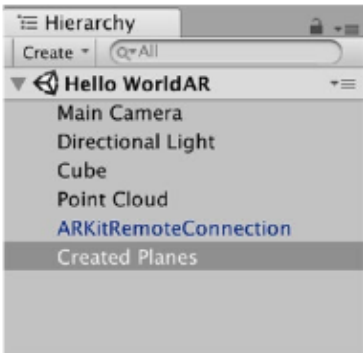
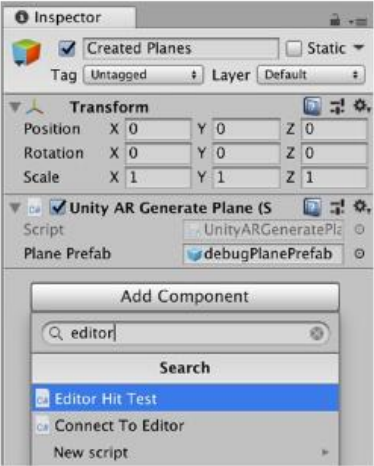
PENGUJIAN TABRAKAN DAN PENCAHAYAAN

Pada bab terakhir, kita menggunakan Cloud Points dan Generated Planes untuk membantu kamera melacak pergerakan perangkat. Namun, Anda mungkin telah memperhatikan bahwa kubus menakutkan kita terus berada di titik asal. Meskipun ini mungkin baik-baik saja untuk Aplikasi AR pertama kita, saya yakin pada waktunya Anda mungkin ingin membuat objek virtual yang tampak berada di objek fisik di dunia nyata.

Untuk melakukannya, kita akan menggunakan fitur Unity ARKit yang disebut pengujian tabrakan. Anda mungkin juga telah memperhatikan bahwa pencahayaan GameObject di layar perangkat iOS Anda tidak konsisten dengan pencahayaan di dunia nyata; kita juga akan membahas masalah ini di bab ini.

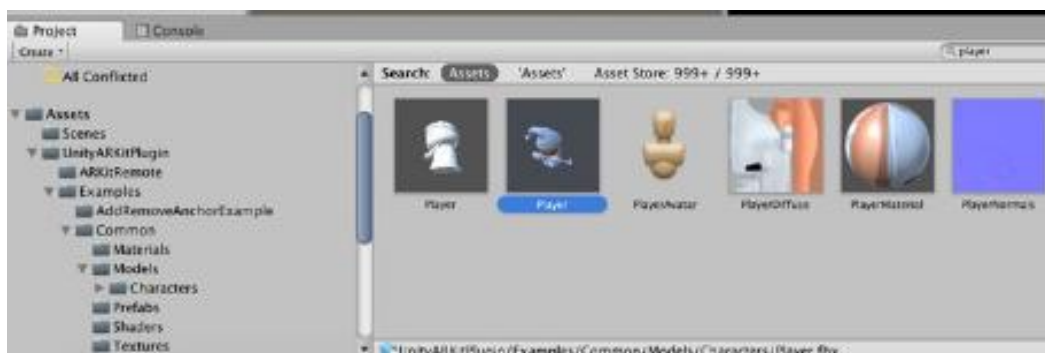
4.1 PENGUJIAN TABRAKAN

Pengujian tabrakan akan memungkinkan kita untuk menempatkan sesuatu di Generated Planes yang kita buat di Bab 3. Ini juga akan memungkinkan Aplikasi AR untuk menempatkan objek agar tampak seperti berada di posisi di dunia nyata yang relatif terhadap tempat pengguna mengetuk layar perangkat mereka. Mari kita buka adegan Hello WorldAR yang kita buat di bab terakhir. Dengan Adegan Hello World AR yang terbuka di Unity, sekarang pilih GameObject Created Planes di Hierarchy (Gambar 4.1). Dengan GameObject Created Planes yang dipilih dalam Hierarchy, pilih Add Component; dan di bilah pencarian, cari Editor Hit test (Gambar 4.2). Sekarang pilih ini untuk menambahkannya ke Game Object Created Planes kita. Skrip ini akan memungkinkan kita untuk menguji fungsionalitas dasar Hit Testing di editor.

	
<p style="text-align: center;">Gambar 4.1 Memilih Gameobject Created Planes Dalam Hierarchy</p>	<p style="text-align: center;">Gambar 4.2 Mencari Editor Hit Test</p>

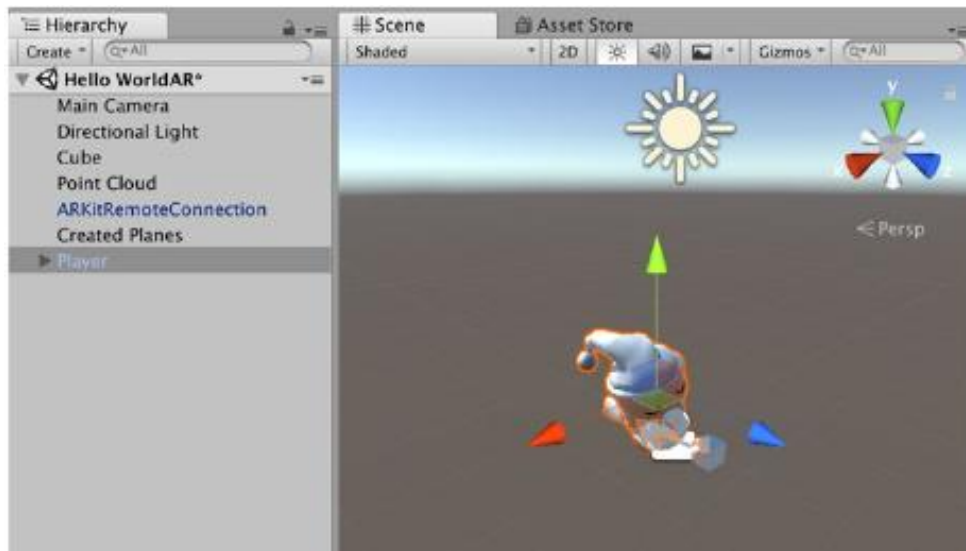
Sekarang Anda mungkin ingin menyertakan sesuatu di Aplikasi AR yang sedikit lebih menarik daripada kubus kita. Kita dapat mencari di Asset Store untuk aset yang menarik atau relevan untuk disertakan, atau kita dapat membuatnya sendiri. Karena membuat aset permainan kita sendiri sangat memakan waktu dan di luar cakupan buku ini, untuk saat ini, mari kita gunakan salah satu aset yang ada di Unity ARKit. Di folder Project, cari aset Player (Gambar 4.3). Jika GameObject ini tidak disertakan dalam versi Unity ARKit yang Anda unduh, Anda dapat mengunduhnya dari Unity Asset Store:

(<https://assetstore.unity.com/packages/essentials/tutorial-projects/survival-shooter-tutorial-40756>).



Gambar 4.3 Mencari Aset Permainan Player

Dengan aset ini dipilih, seret ke Hierarchy (Gambar 4.4).



Gambar 4.4 Karakter Player Ditambahkan Ke Hierarchy

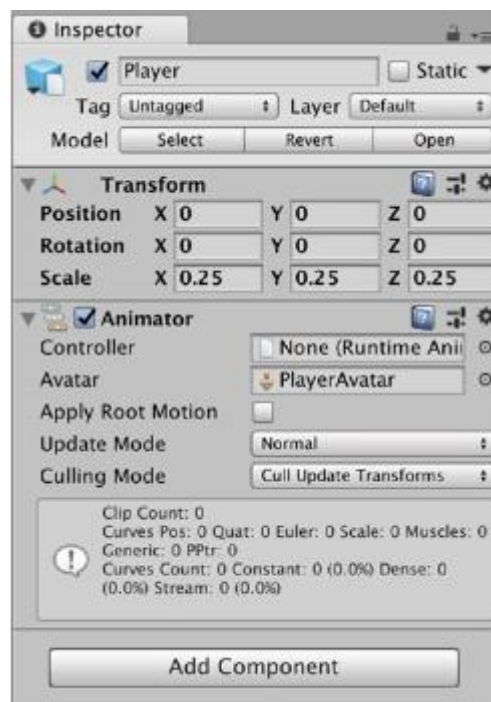
Sekarang saatnya mengucapkan selamat tinggal pada kubus menakjubkan kita. Pilih Kubus di Hierarki dan klik kanan lalu pilih hapus (Gambar 4.5).



Gambar 4.5 Menghapus Kubus

4.2 SKALA

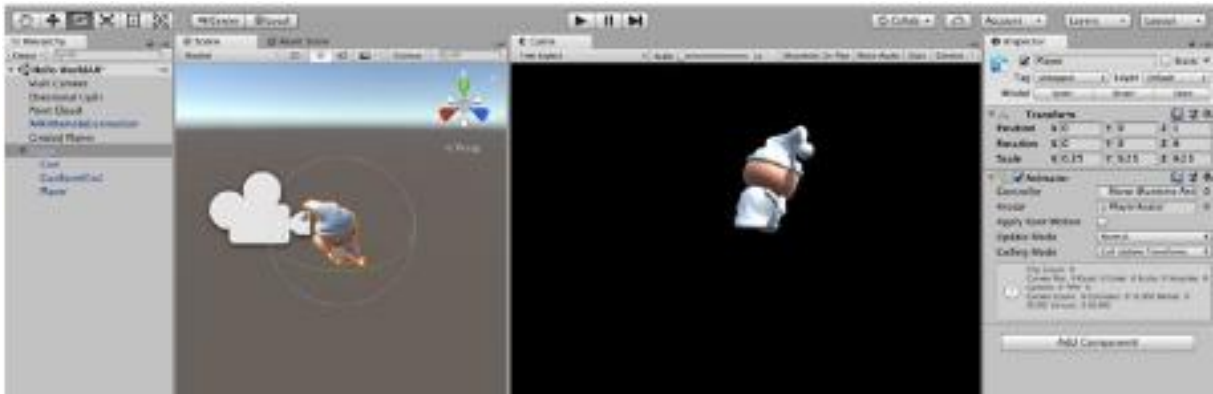
Anda mungkin ingat bahwa di Bab 2, saya membahas pentingnya skala dalam AR. Jika Anda memilih GameObject Pemain di Hierarki, Anda akan melihat bahwa aset ini tingginya 1 meter. Mungkin ini bagus, tetapi saya ingin membuat aset ini terlihat jauh lebih kecil di dunia nyata. Saya menduga bahwa ini adalah maksud dari orang yang membuat aset ini. Dengan GameObject Pemain yang dipilih, ubah skala menjadi .25,.25,.25 (Gambar 4.6).



Gambar 4.6 GameObject Pemain yang Diubah Skalanya

4.3 TRANSFORMASI

Jika Anda memilih Kamera Utama dan melihat jendela Pratinjau Kamera, Anda akan melihat bahwa GameObject Pemain kita tidak terlihat oleh pemain. Jika Anda melihat Inspektur atau Kamera Utama dan GameObject Pemain, Anda akan melihat bahwa posisi kedua aset ini adalah 0,0,0. Mari kita pindahkan GameObject Pemain. Dengan Pemain, GameObject dipilih, di Inspektur ubah posisi menjadi 0,0,1 (Gambar 4.7).



Gambar 4.7 GameObject Pemain Diubah Posisinya

Anda akan melihat pada Gambar 4.7 bahwa GameObject Pemain menghadap menjauh dari kamera. Saya pikir akan lebih baik jika GameObject Pemain menghadap pemain. Dengan GameObject Pemain dipilih, gunakan alat putar, dan putar GameObject Pemain pada sumbu Y hingga menghadap Kamera utama (Gambar 4.8).



Gambar 4.8 Memutar GameObject Pemain

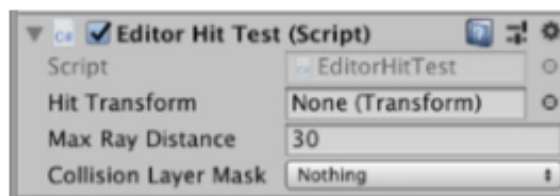
Saat memutar GameObject, ada kemungkinan untuk mengubah sumbu lain secara tidak sengaja. Pada Gambar 4.8, perhatikan bagaimana posisi Transformasi sumbu X dan sumbu Y juga telah berubah. Mari kita atur ulang posisi sumbu X ke 0 dan sumbu Z ke 1. Saya lebih suka bilangan bulat, jadi saya akan mengatur rotasi sumbu Y ke 156 (Gambar 4.9).



Gambar 4.9 GameObject Pemain yang diputar

4.4 SKRIP UJI TABRAKAN EDITOR

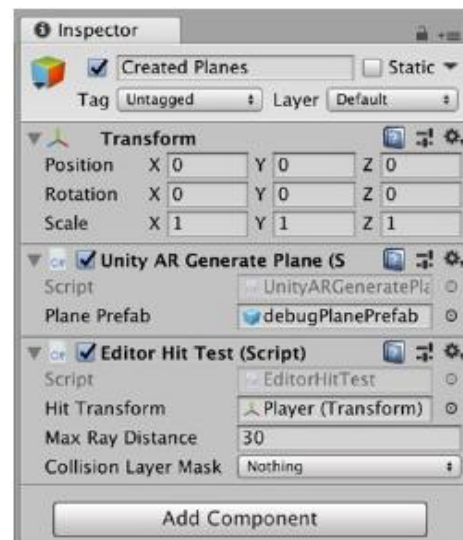
Sekarang pilih skrip Uji Tabrakan Editor yang ada di GameObject Bidang yang Dibuat. Skrip Uji Tabrakan Editor memiliki tiga parameter yang dapat diubah: Transformasi Tabrakan, Jarak Sinar Maksimum, dan Lapisan Masker Tabrakan (Gambar 4.10).



Gambar 4.10 Skrip Uji Hit Editor

Transformasi Hit

Kotak parameter Transformasi Hit menentukan transformasi GameObject. Dalam contoh kami, ini akan menjadi x, y, z dari GameObject Pemain yang akan dipindahkan ke tempat yang telah dipilih pemain. Skrip uji Hit Editor saat ini tidak mendukung ketuk atau sentuh, jadi dalam kasus ini, GameObject akan diposisikan di tempat pemain mengklik di tab Game dari Editor Unity. Pilih GameObject Pemain di Hierarki dan seret ini ke kotak properti Transformasi Hit di skrip Uji Editor Hit (Gambar 4.11).



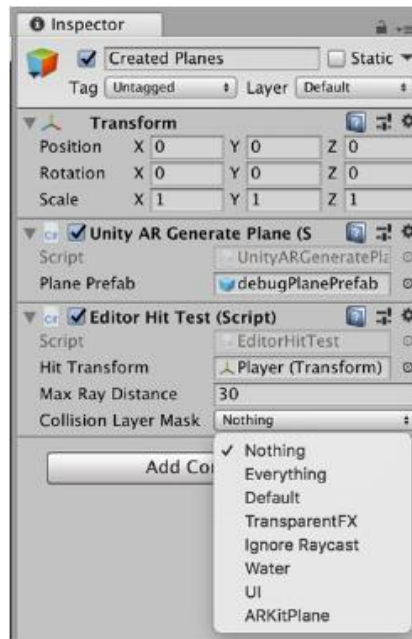
Gambar 4.11 Set Transformasi Gameobject Pemain Dalam Properti Hit Transform

Jarak Sinar Maksimum

Untuk latihan ini, kita tidak akan mengubah Jarak Sinar Maksimum. Jika Anda tertarik mempelajari parameter ini, ada banyak informasi tentang Ray Casting di Internet. Singkatnya, ini adalah proses memancarkan sinar tak kasatmata untuk mendeteksi jika ada sesuatu (seperti collider) di jalur sinar tersebut.

Topeng Lapisan Tabrakan

Menu tarik-turun Topeng Lapisan Tabrakan memiliki beberapa opsi (Gambar 4.12).

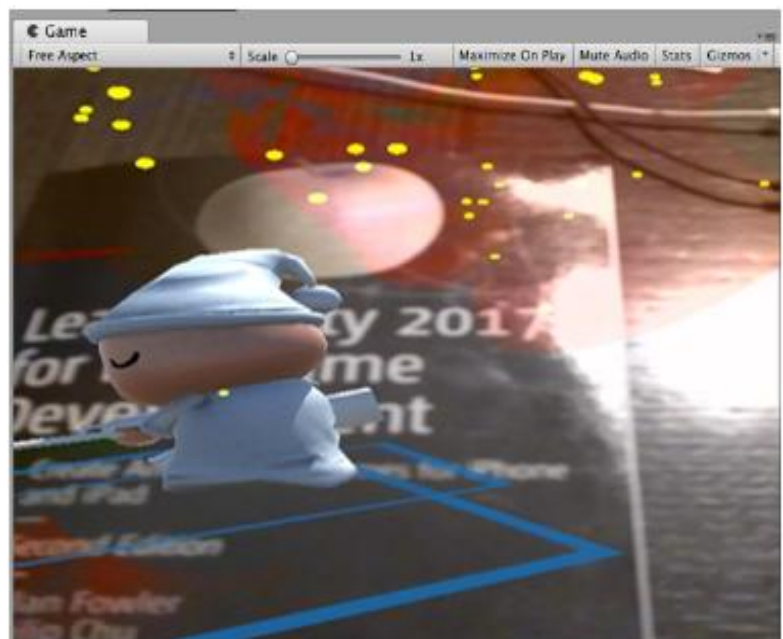


Gambar 4.12 Opsi Collision Layer Mask

Kita akan memilih opsi ARKitPlane. Opsi Collision Layer Mask mengatur lapisan masker tabrakan yang akan dipertimbangkan selama pengujian benturan. Dalam kasus kita, kita akan memilih ARKitPlane.

Pengujian

Sekarang hubungkan perangkat iOS Anda ke Mac dan jalankan Aplikasi UnityARKitRemote di perangkat Anda. Di Konsol di Unity, pilih perangkat Anda dari Editor, dan tekan tombol Putar. Di Layar Game, klik ikon Start ARKitRemote, lalu gerakkan perangkat perlahan agar kamera menemukan bidang yang sesuai. Saat Anda melihat bidang di tab Game di Unity, klik di dalam bidang itu, dan Anda akan melihat avatar Pemain Anda di layar tampilan Game (Gambar 4.13).



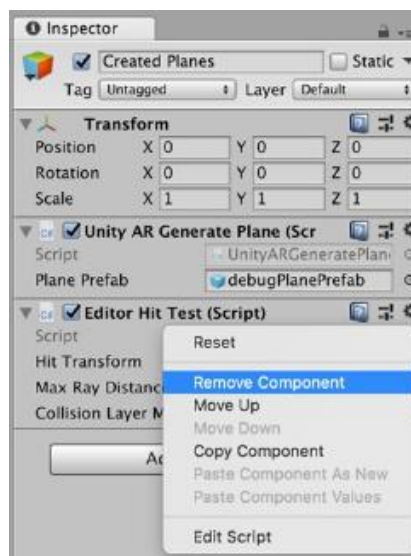
Gambar 4.13 GameObject Pemain di layar tampilan Game

Pengujian di Perangkat Kita

Sekarang setelah kita berhasil menguji Aplikasi AR kita di editor, saatnya untuk melihat seperti apa tampilannya di perangkat kita. Anda mungkin ingat selama pengujian build, kita menambahkan skrip Editor Hit Testing. Skrip ini sangat berguna untuk menguji Aplikasi kita di editor, tetapi seperti yang kita temukan, skrip ini tidak mendukung interaksi pengguna dengan mengetuk atau menyentuh di perangkat kita. Jika kita ingin menguji Aplikasi di perangkat kita, ada opsi yang lebih baik.

Menghapus Komponen

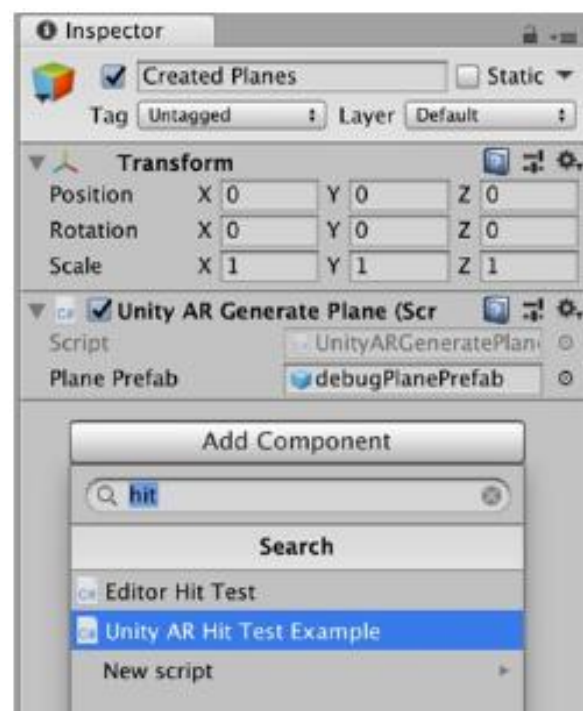
Untuk menghapus komponen skrip Editor Hit Testing dari GameObject Bidang yang Dibuat, kita perlu memilih Roda Gigi di sebelah kanan (Gambar 4.14). Pilih opsi Remove Component.



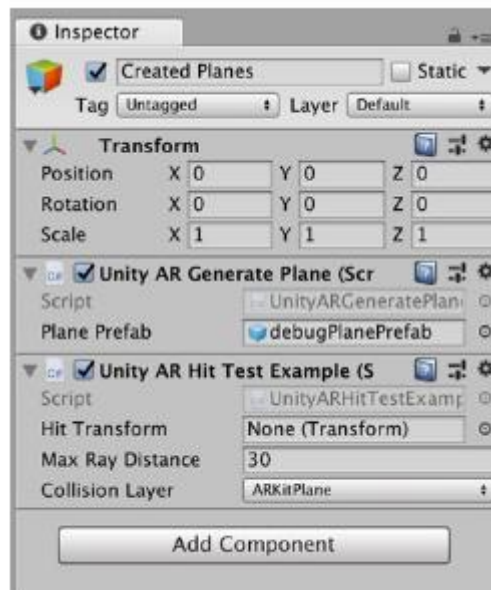
Gambar 4.14 Opsi Menu Remove Component

4.5 Menambahkan Komponen

Sekarang mari tambahkan komponen yang akan mendukung interaksi pengguna yang berfungsi penuh pada perangkat kita. Di Inspector GameObject Created Planes kita, pilih tombol Add Component dan di bilah pencarian cari hit (Gambar 4.15). Sekarang pilih Unity AR Hit Test Example Script untuk menambahkannya ke GameObject Created Plane kita (4.16).



Gambar 4.15 Mencari Contoh Uji Coba AR Unity



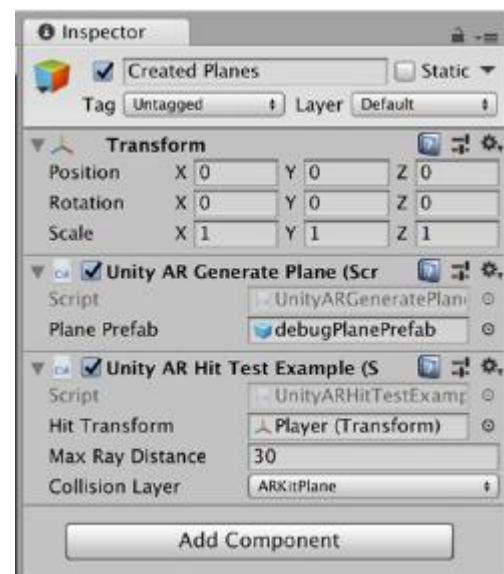
Gambar 4.16 Skrip Contoh Uji Coba AR Unity Yang Ditambahkan Ke Gameobject Generated Planes

Menambahkan Hit Transform

Seperti yang kita lakukan dengan Contoh Uji Hit Editor, kita perlu menambahkan parameter Hit Transform. Untuk melakukannya, pilih GameObject Pemain dan seret ke kotak properti Hit Transform di Contoh Uji AR Kit Unity (Gambar 4.17). Jika ARKitPlane tidak ditetapkan sebagai Collision Layer, gunakan menu tarik-turun, ubah ini ke ARKitPlane.

Persiapan untuk Penerapan

Sebelum kita membangun dan menjalankan Aplikasi baru, kita perlu mengubah pengaturan Build. Dari menu, pilih File ► Build Settings (Gambar 4.18).



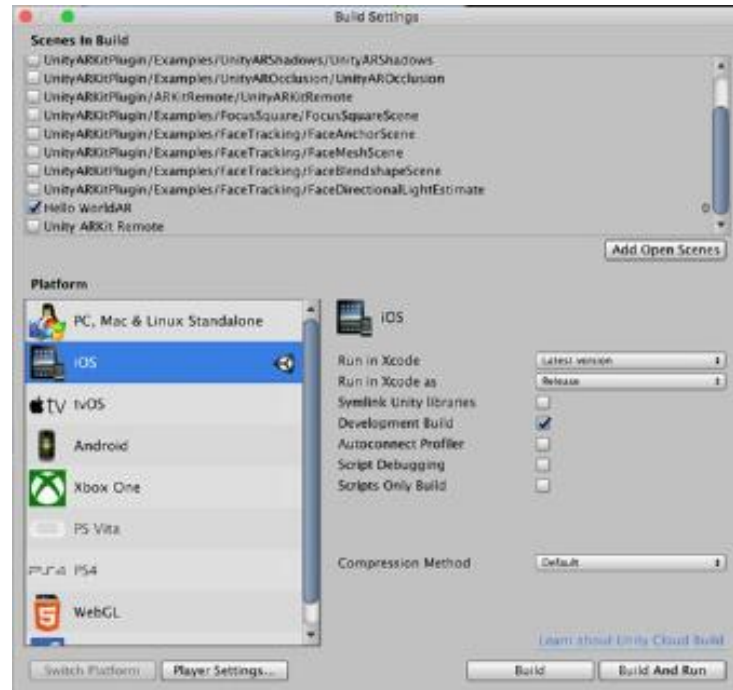
Gambar 4.17 Transformasi Pemain Yang Ditambahkan Ke Properti Transformasi Hit



Gambar 4.18 Pilih Pengaturan Bangun

Mengubah Pengaturan Bangun

Di menu Pengaturan Bangun, periksa apakah Anda sedang membangun adegan saat ini (Hello WorldAR) di opsi Adegan untuk Dibangun (Gambar 4.19).



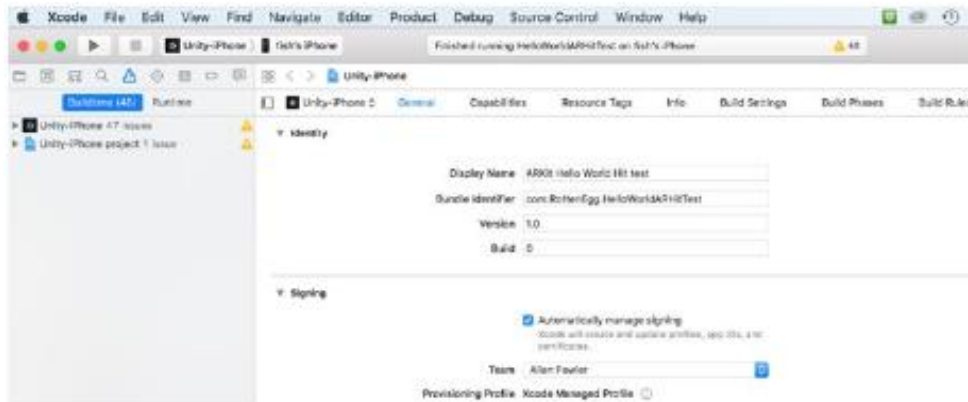
Gambar 4.19 Memilih Opsi Scenes To Build

Pilih Player Settings dan di Inspector, ubah Product Name dan Bundle Identifier (Gambar 4.20).



Gambar 4.20 PlayerSettings Inspector

Sebelum memilih Build and Run, saya sarankan untuk membuka Xcode terlebih dahulu dan memastikan bahwa kita telah memilih perangkat target dan telah mengatur Team di menu Signing (Gambar 4.21). Jika Anda tidak memilih ini sekarang, Anda akan menemukan pesan build failed. Sekarang di menu Build Settings, pilih Build and Run.



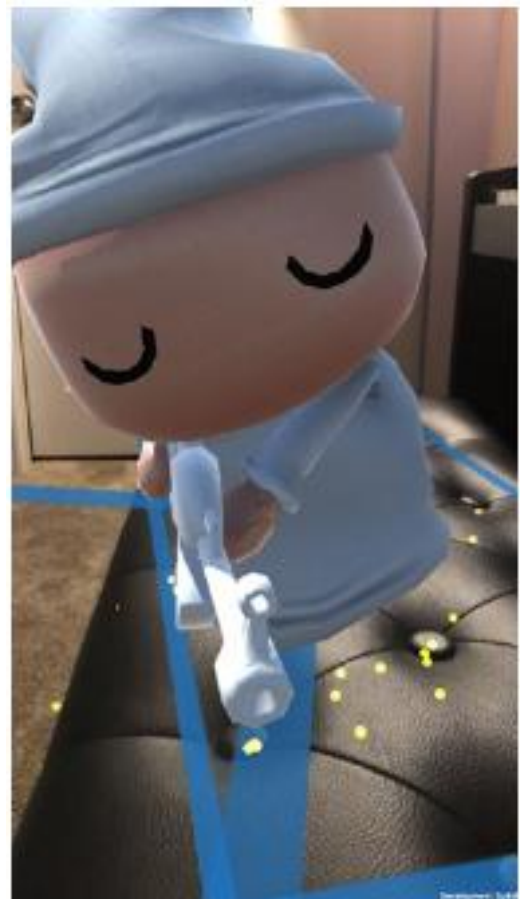
Gambar 4.21 Kotak Pilihan Tim Penandatanganan Menu Xcode

Setelah Anda berhasil membuat dan menerapkan Aplikasi ke perangkat iOS Anda, Anda seharusnya dapat menjalankan Aplikasi tersebut; dan saat Anda mengetuk layar pada perangkat Anda, Anda akan melihat GameObject Pemain di layar Anda ditempatkan pada posisi di dunia nyata yang relatif terhadap posisi tempat Anda mengetuk layar (Gambar 4.22).

Meskipun GameObject Pemain kita jauh lebih menarik daripada kubus kita, kita dapat meningkatkan Aplikasi ini. Jika kita melihat pencahayaan GameObject Pemain, pencahayaannya tidak sepenuhnya konsisten dengan pencahayaan di dunia nyata. Kita akan mengubah pencahayaan gambar AR, sehingga lebih mendekati pencahayaan di dunia nyata. Anak-anak saya berpikir kita harus mengganti nama GameObject Pemain kita menjadi Nightmare Chaser, jadi mari kita ganti nama GameObject Pemain kita menjadi Nightmare Chaser juga.

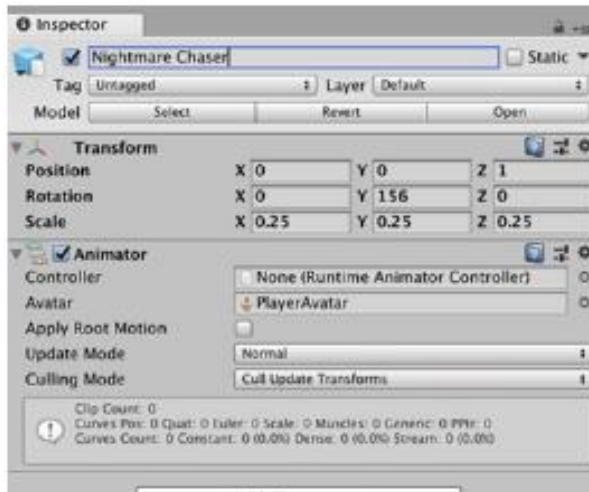
Mengubah Nama GameObject Pemain

Pengubahan nama GameObject Pemain dapat dilakukan dengan memilih GameObject ini di Hierarchy, lalu di Inspector, pilih nama saat ini dan ketik nama baru kita (Gambar 4.23). Pilihan lainnya adalah Klik kanan GameObject Pemain di Hierarchy dan pilih opsi Rename dari menu

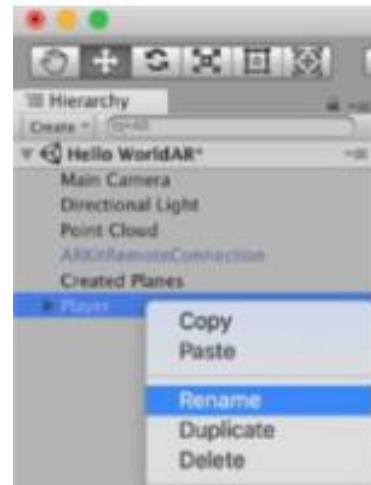


Gambar 4.22 Aplikasi Hello WorldAR yang diterapkan pada iPhone

drop-down (Gambar 4.24).



Gambar 4.23 Mengubah nama GameObject Pemain di Inspektur



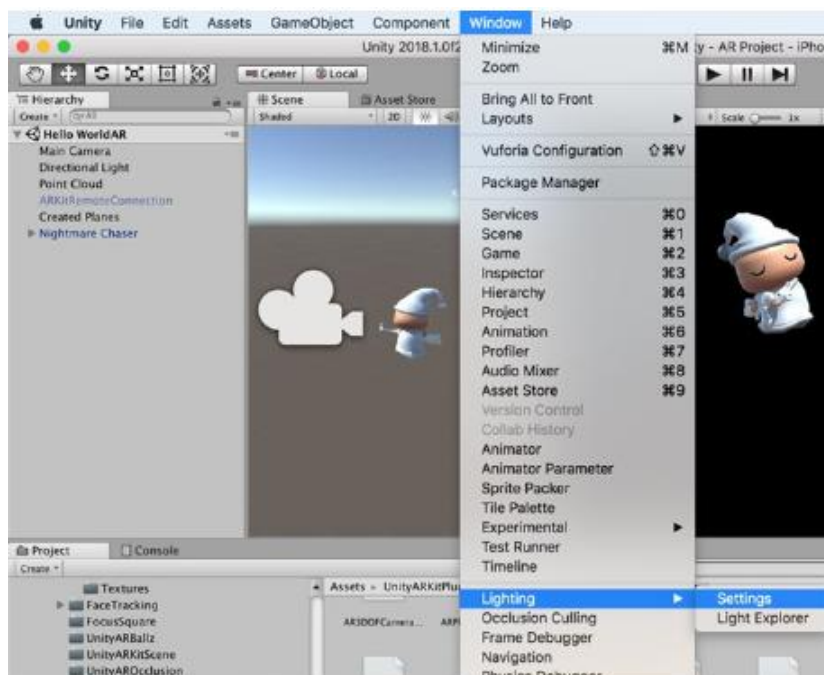
Gambar 4.24 Mengubah nama GameObject Pemain di Hirarki

4.6 PENCAHAYAAN

Beberapa dari Anda mungkin telah memperhatikan bahwa pencahayaan Nightmare Chaser tidak konsisten dengan pencahayaan di dunia nyata. Sekali lagi, orang-orang hebat di Unity telah memberi kita solusi (sebagai catatan, saya tidak berafiliasi dengan Unity; saya pikir mereka melakukan pekerjaan yang hebat pada Unity ARKit).

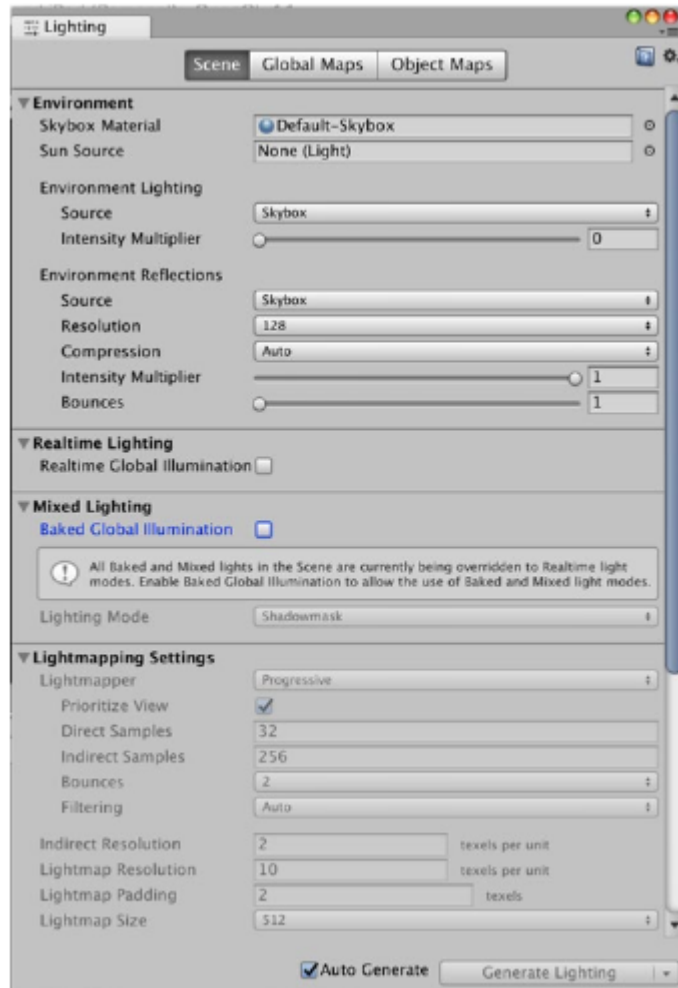
Mematikan Lampu

Kita akan menggunakan sumber cahaya yang berbeda untuk Aplikasi kita. Namun, untuk hasil terbaik, mari kita pastikan kita telah mematikan semua lampu di tempat kejadian. Dari Menu Utama, pilih Jendela ► Pencahayaan ► Pengaturan (Gambar 4.25).



Gambar 4.25 Memilih Pengaturan Pencahayaan

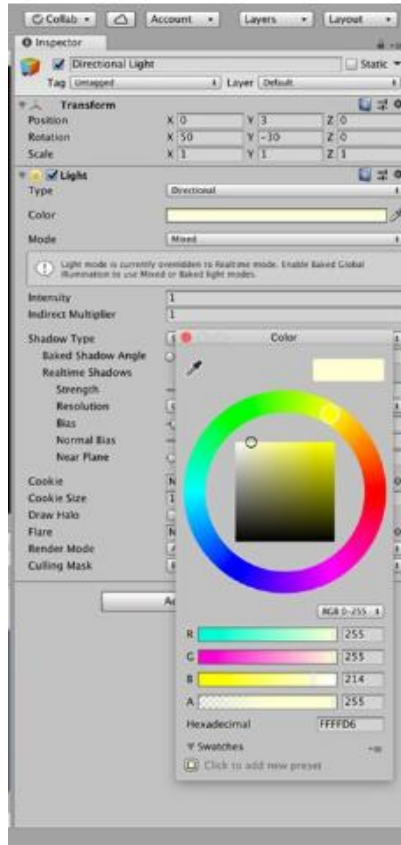
Pada Pengaturan Pencahayaan, pada pengaturan Lingkungan ubah Pengganda Intensitas menjadi nol. Pada pengaturan Pencahayaan Campuran, hapus centang pada pengaturan Pencahayaan Terpanggang (Gambar 4.26).



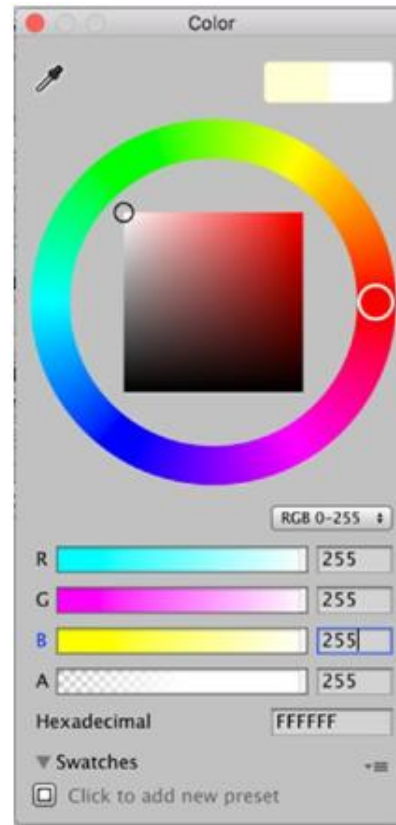
Gambar 4.26 Mengubah Pengaturan Pencahayaan

Sekarang pilih Directional Light dan di Inspector, ubah warnanya menjadi putih. Ada sejumlah cara untuk mengubah warna. Pertama di Inspector, pilih kotak properti warna dan pilih warna akan muncul (Gambar 4.27). Di menu pemilih Color, ada roda pemilih warna untuk memilih warna. Untuk menggunakan roda ini, pilih dan seret lingkaran pemilihan warna kecil hingga Anda menemukan warna yang mendekati warna yang Anda inginkan. Di tengah lingkaran, ada kotak untuk memilih rona.

Setelah Anda memilih warna yang mendekati warna yang Anda inginkan, Anda dapat menggunakan lingkaran kecil di kotak tersebut untuk menyempurnakan pilihan Anda. Metode ini mungkin berhasil bagi sebagian orang, tetapi saya cukup menyetik nilai RGB atau HSV dari warna yang saya inginkan di kotak properti (Gambar 4.28).



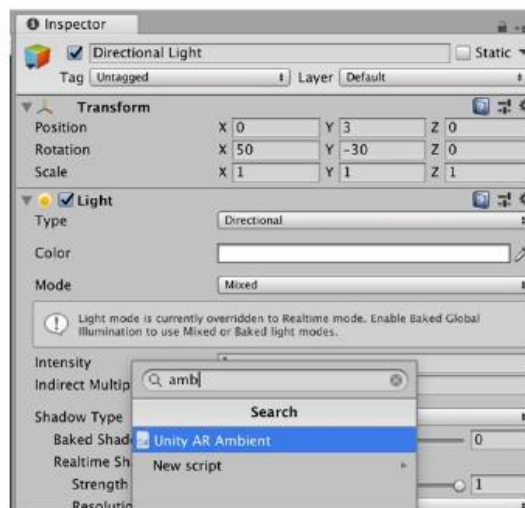
Gambar 4.27 Menggunakan Roda Warna



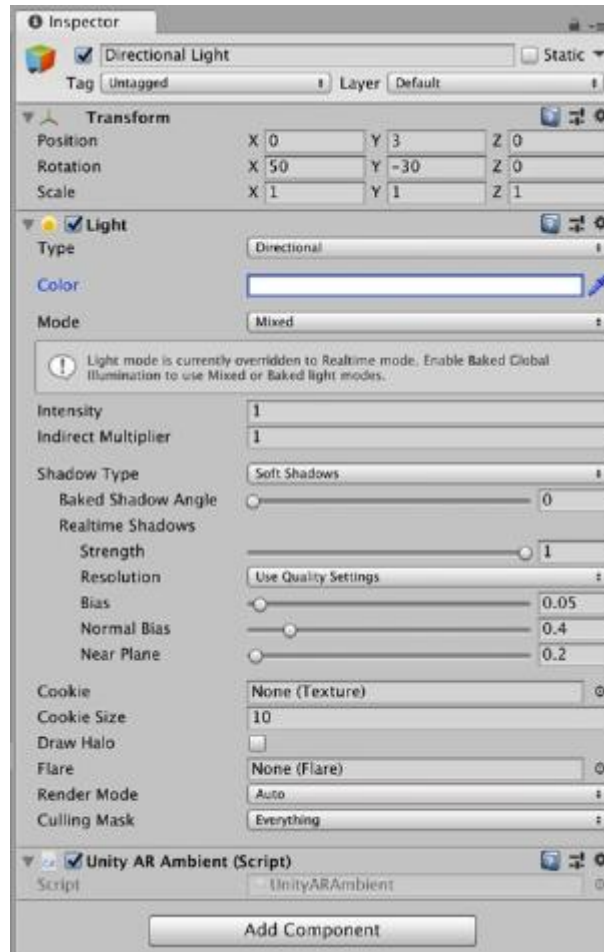
Gambar 4.28 Mengetik Di Properti Warna

Mengatur Sumber Cahaya Sekitar

Sekarang kita dapat mengatur sumber cahaya sekitar. Dengan memilih Cahaya Terarah, pilih tombol Tambahkan Komponen dan cari ambient (Gambar 4.29).



Gambar 4.29 Mencari Komponen Ambient Unity AR



Gambar 4.30 Skrip Unity AR Ambient Yang Ditambahkan Ke Directional Light

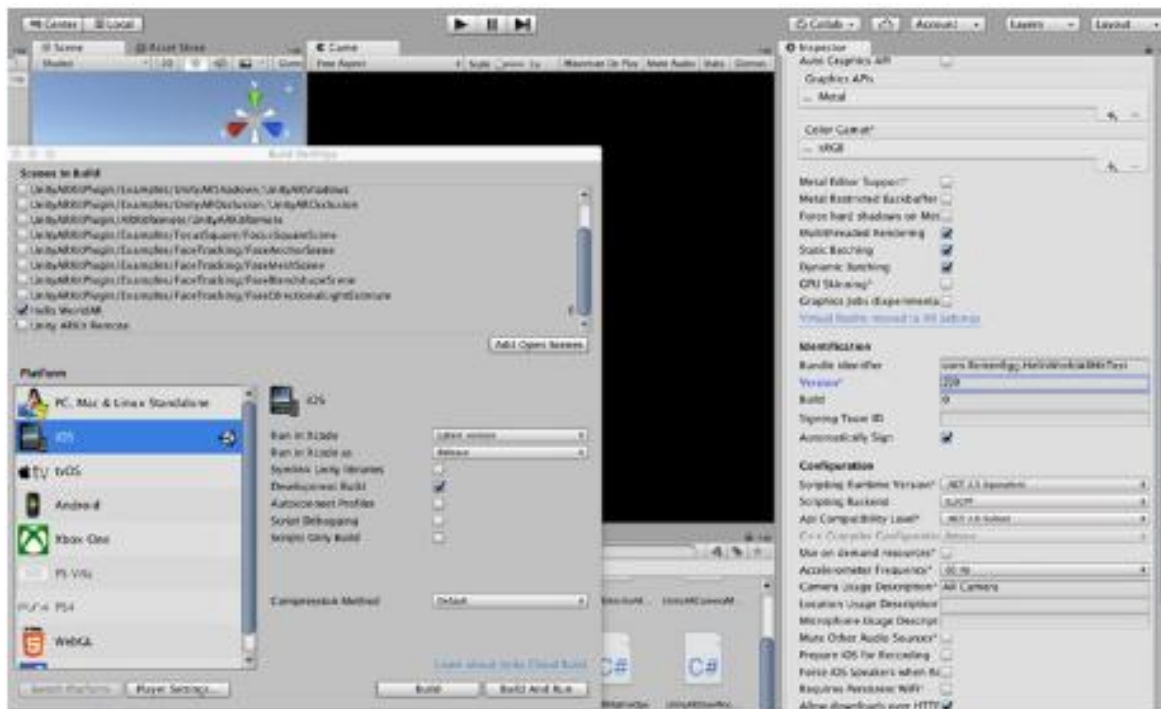
Anda akan melihat bahwa Unity ARKit juga menyertakan skrip yang disebut Unity AR Ambient. Pilih skrip ini untuk menambahkannya ke lampu terarah kita (Gambar 4.30). Skrip ini memperkirakan pencahayaan dunia nyata dan akan membuat perubahan pada pencahayaan di Aplikasi kita. Ini akan membuat Aplikasi kita terlihat lebih realistis. Sekarang mari kita uji aplikasi ini dalam kondisi pencahayaan yang berbeda. Sekarang, Anda seharusnya sudah tahu cara membuat dan menjalankan adegan. Sebelum kita melakukan ini, saya sarankan untuk membuat satu perubahan kecil di Pengaturan Pemain.

Build and Run – Version Control

Saya penggemar berat kontrol versi. Menurut saya, menyimpan pekerjaan dengan nomor versi yang berbeda saat membuat perubahan besar atau kecil adalah praktik yang baik. Bahkan, saat menulis buku ini, saya telah menggunakan kontrol versi. Saya telah mengalami terlalu banyak kejadian sistem saya mogok dan tidak dapat memulihkan versi terakhir dari proyek yang sedang saya kerjakan. Untungnya, sering kali, saya memiliki versi sebelumnya yang dapat saya gunakan dan kembali bekerja dengan relatif cepat.

Karena kami telah memiliki versi sebelumnya dari uji Hit AR Hello World kami, saya akan menyimpan pengujian berikutnya ini dengan nomor versi yang berbeda. Pilih Build Settings dari menu utama, lalu pilih Player Settings (Gambar 4.31). Di Inspector pengaturan Player, gulir ke bawah untuk melihat pengaturan Identification, dan di bawah Version ubah

versi menjadi 2.0. Beberapa orang mungkin berpendapat bahwa versinya harus 1.1 (karena ini hanya perubahan kecil), tetapi selama kita mengubah nomor versi, itulah yang paling penting.



Gambar 4.31 Mengubah Identifikasi Versi di tampilan Inspector

Sekarang buka Xcode, pastikan Anda sudah masuk, dan hubungkan perangkat iOS Anda ke Mac dan pilih Build and Run. Jika build Anda berhasil, Anda akan melihat pencahayaan di Aplikasi berubah dalam kondisi pencahayaan yang berbeda.

Memposisikan Kamera

Anda mungkin telah menemukan bahwa meskipun kami dengan hati-hati memposisikan Nightmare Chaser kami sehingga ia menghadap kamera pemain, ia tidak selalu menghadap kamera saat kami menjalankan Aplikasi di perangkat kami. Posisi Nightmare Chaser akan bergantung pada posisi perangkat iOS Anda saat Anda memulai Aplikasi. Meskipun ini adalah buku untuk pemula, kita harus menulis beberapa kode. Semoga ini tidak terlalu menyulitkan Anda. Kode ini akan secara otomatis memutar Nightmare Chaser kami untuk menghadap kamera setiap kali kita memulai Aplikasi.

Mengedit Skrip Uji Tabrakan Unity AR

Di Hierarchy, pilih GameObject Created Planes; dan di Inspector, pilih UnityARHitTestExample di kotak Script Properties. Ini akan menemukan dan menyorot skrip UnityARHitTestExample.cs di Folder Proyek (Gambar 4.32).



Gambar 4.32 Memilih Skrip Unityarhittestexample.Cs

Jika Anda tidak memiliki salinan Visual Studio atau Mono Develop, saya sangat menyarankan untuk mengunduh salinan dan menginstalnya sekarang. Meskipun kita tidak akan menulis banyak kode di buku ini, jika Anda serius ingin mengembangkan game AR untuk iOS, maka Anda memerlukan IDE. Dengan skrip UnityARHitTestExample.cs yang dipilih, klik dua kali file ini, dan skrip tersebut akan terbuka di IDE Anda (Visual Studio atau MonoDevelop). Pada Baris 18, masukkan kode berikut:

```
//Automatically Face the Camera
m_HitTransform.LookAt (Camera.main.transform.position);
m_HitTransform.eulerAngles = new Vector3 (0, m_HitTransform.
eulerAngles.y, 0);
```

Sekarang simpan skrip dan uji untuk melihat apakah ini berjalan. // menambahkan komentar ke kode dan diabaikan saat dikompilasi. Jika Anda menemukan kesalahan runtime atau bug dengan kode, cukup komentari tiga baris ini, dan kode akan berjalan (meskipun, Nightmare Chaser kita tidak akan secara otomatis diposisikan sehingga menghadap kamera).

Ringkasan

Dalam bab ini, kita telah mempelajari tentang Hit Testing, yang membantu Aplikasi kita mengidentifikasi permukaan di dunia nyata dan "menempatkan" GameObject pada permukaan tersebut. Kita juga mempelajari cara mengubah pencahayaan sehingga konsisten dengan pencahayaan di dunia nyata.

BAB 5

MEMBUAT GAME AR

Sekarang setelah Anda mempelajari beberapa dasar penggunaan ARKit dan memahami Unity, kita akan mulai membuat Game AR. Di sepanjang jalan, kita juga akan membahas beberapa fungsi ARKit yang lebih canggih.

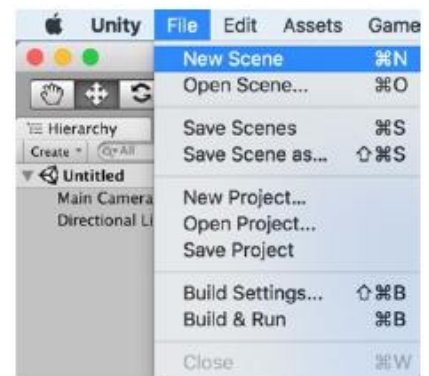
5.1 FUGU BOWLGAME AR

Sekarang saatnya membuat Game AR. Saya telah memutuskan untuk menghidupkan kembali game yang kita buat di buku terakhir kita, *Learn Unity for iOS Game Development* (Apress, 2017). Jika Anda salah satu dari sedikit orang yang tidak mendapatkan buku ini, game kita adalah Fugu Bowl, yang merupakan game bowling kecil untuk perangkat iOS. Di buku terakhir kita, kita menggunakan JavaScript, tetapi untuk versi AR, kita akan menggunakan C#. Meskipun JavaScript adalah bahasa pemrograman yang hebat, jika Anda akan terus menggunakan Unity, maka mempelajari C# akan jauh lebih penting dan berharga.

Jika Anda tidak tahu cara memprogram dalam C#, saya sangat menyarankan untuk mempelajari bahasa ini. Buku ini berfokus pada pembelajaran membuat Game AR, jadi saya tidak akan fokus mengajarkan cara membuat program dalam C#. Namun, saya telah menyertakan komentar dalam kode untuk membantu siapa pun yang tidak familier dengan C# memahami apa yang akan dilakukan kode tersebut.

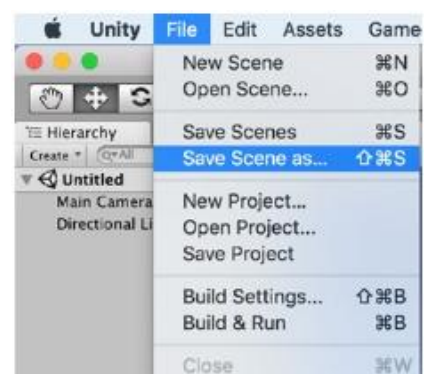
Membuat Adegan Baru

Karena kita telah mengunduh dan memasang paket Unity ARKit ke dalam Proyek AR kita, mari kita lanjutkan menggunakannya. Biasanya, saya lebih suka (dan merekomendasikan) memulai Proyek baru untuk setiap permainan baru yang dibuat. Namun, karena ini akan melibatkan pengunduhan paket Unity ARKit, mari kita lewatkan saja kali ini. Untuk membuat Adegan Baru, dari menu utama, pilih File ► Adegan Baru (Gambar 5.1).



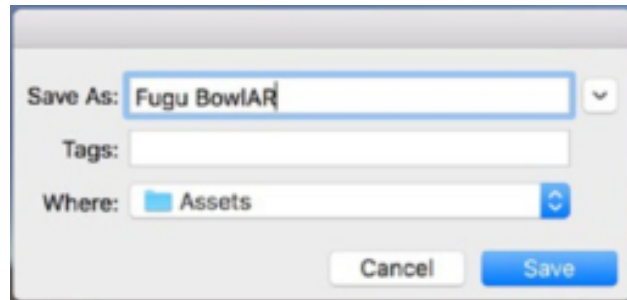
Gambar 5.1 Membuat Adegan Baru

Mungkin lebih efisien untuk menggunakan tombol pintas (⌘-N), dan saya juga akan menyertakan tombol pintas untuk perintah yang sering digunakan bagi pembaca yang lebih suka menggunakannya. Mari Simpan Adegan Kita. Pilih dari menu utama, File ► Simpan Adegan sebagai (Gambar 5.2).



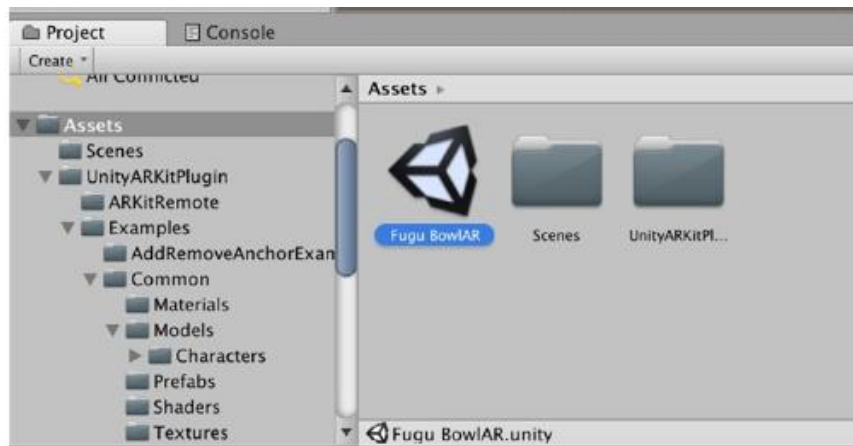
Gambar 5.2 Simpan Adegan

Anda akan disajikan dengan kotak dialog Simpan Sebagai (Gambar 5.3) dan ketik nama untuk Adegan kita. Saya telah memutuskan untuk memberi nama permainan ini Fugu BowlAR (lihat apa yang saya lakukan di sana?).

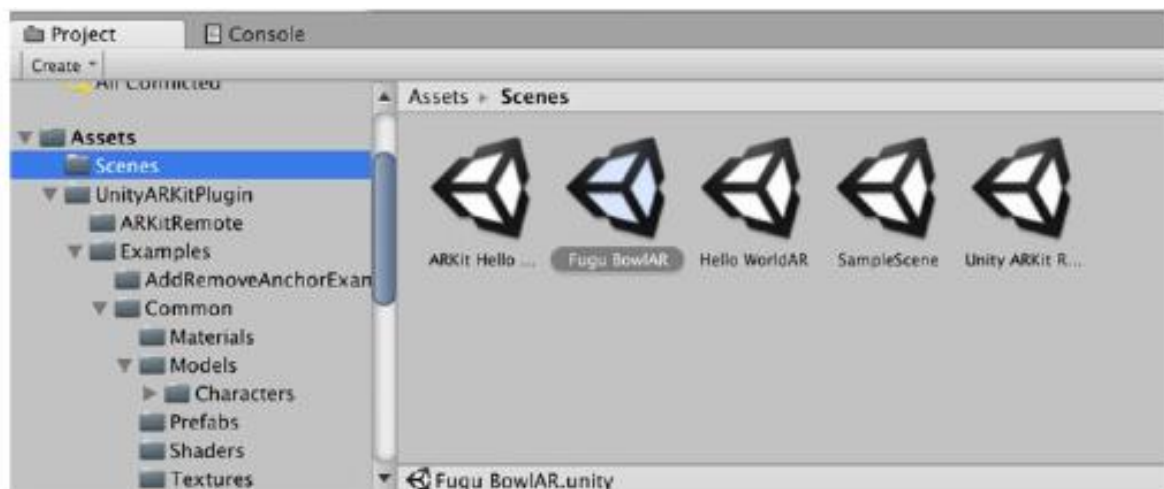


Gambar 5.3 Menyimpan Adegan Sebagai Fugu Bowlar

Pilih Simpan untuk menyimpan Adegan ini. Anda akan melihat bahwa berkas ini sekarang disimpan dalam folder Aset. Saya percaya bahwa dalam pengembangan perangkat lunak, lebih baik untuk terorganisasi dengan baik (beberapa orang mungkin menyebut ini obsesif). Oleh karena itu, Adegan ini mungkin harus disimpan dalam folder Adegan. Dalam folder proyek, pilih adegan Fugu BowlAR (Gambar 5.4) dan seret ke folder Adegan (Gambar 5.5).



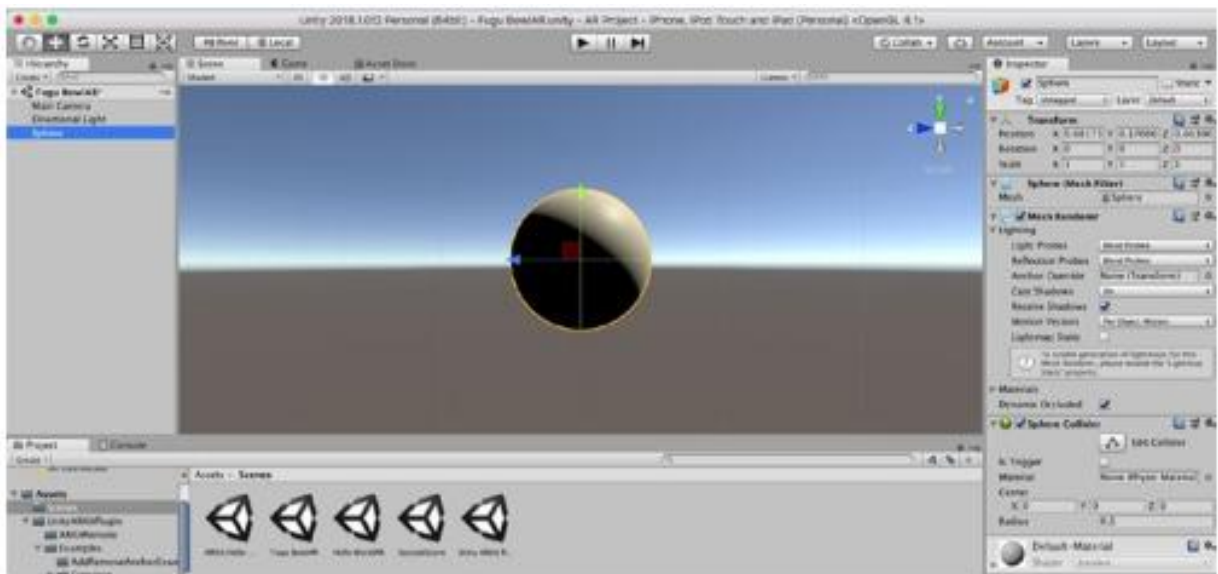
Gambar 5-4 Memilih Fugu Bowlar Scene Di Folder Project Assets



Gambar 5.5 Fugu Bowlar Di Folder Scenes Project

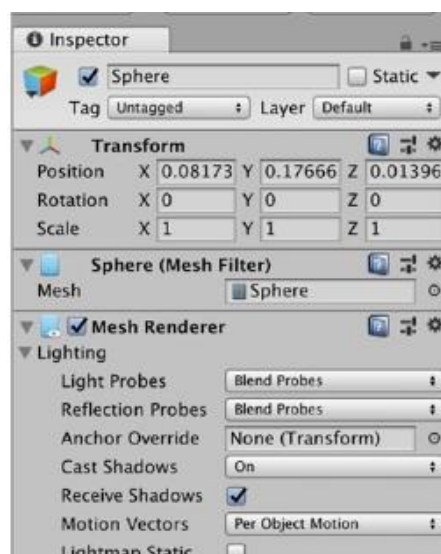
5.2 MEMBUAT ASET AR

Anda mungkin ingat membuat kubus yang menakjubkan, dan kita akan mengulangi proses ini. Pertama, kita akan membuat bola bowling yang menakjubkan. Kita mungkin bisa menemukan bola bowling yang benar-benar menakjubkan secara daring atau mungkin di Unity Asset Store, tetapi saya sangat suka membuat aset game saya sendiri ketika saya punya waktu dan tidak punya anggaran untuk membayar seseorang dengan bakat artistik yang lebih dari saya untuk melakukannya untuk saya (bakat artistik saya sangat terbatas). Di menu utama, pilih GameObject ► Objek 3D ► Bola (Gambar 5.6).

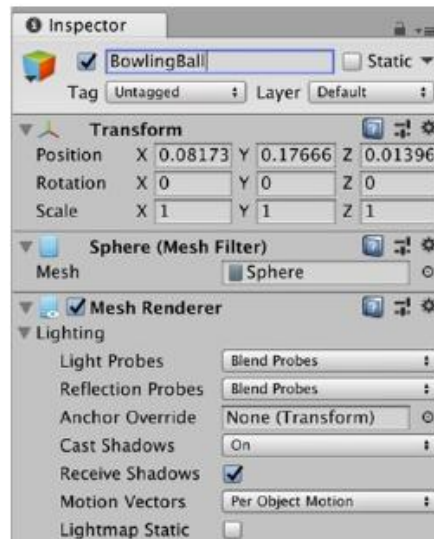


Gambar 5.6 GameObject Sphere Di Scene

Mari kita ubah beberapa properti. Tugas pertama adalah mengganti nama GameObject ini. Saya akan memberi nama GameObject ini BowlingBall. Dengan GameObject Sphere yang dipilih, di Inspector, pilih nama GameObject (Gambar 5.7) dan pilih teks saat ini dan ketik nama baru kita (Gambar 5.8).



Gambar 5.7 GameObject Sphere yang dipilih



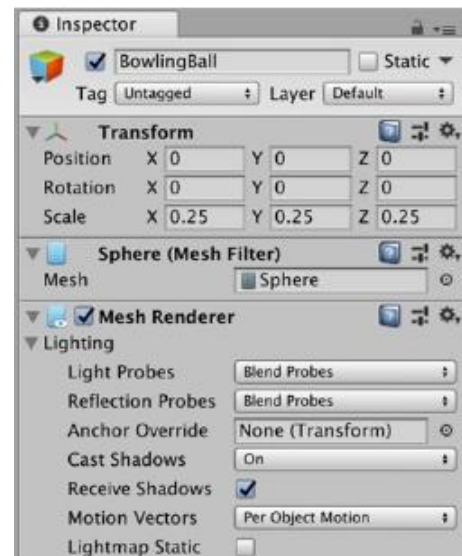
Gambar 5.8 GameObject Sphere yang diubah namanya menjadi BowlingBall

Ubah BowlingBall

Anda mungkin ingat bahwa di Unity kita bekerja dalam metrik. Ini berarti bahwa GameObject BowlingBall kita akan berukuran 1 meter, kali 1 meter, kali 1 meter. Saya belum menghabiskan banyak waktu bermain bowling, tetapi saya pun tahu bahwa ini akan menjadi bola bowling yang sangat besar. Mari kita ubah ukurannya (skalakan) dan letakkan BowlingBall di titik asal (0,0,0). Pertama, pilih Skala dan ketik dalam pengaturan Skala berikut 0,25,0,25,0,25. Sekarang mari kita atur Posisi Transformasi ke 0,0,0 (Gambar 5.9).

Menambahkan Rigidbody

Saat kita membuat kubus, kubus itu melayang di udara, yang cukup keren. Namun karena kita ingin bola itu menggelinding di lantai di dunia nyata, kita perlu menambahkan komponen Rigidbody. Komponen Rigidbody menyediakan simulasi rigid body yang mensimulasikan bagaimana bentuk yang tidak berubah bereaksi terhadap gaya dan benturan. Kita ingin GameObject BowlingBall kita bereaksi terhadap gravitasi (tetap di lantai, dan kita juga akan menerapkan beberapa gaya padanya untuk membuatnya bergerak). Terakhir, kita ingin bola itu bertabrakan dengan pin bowling kita. Jadi, kita perlu menambahkan komponen Rigidbody ke BowlingBall kita. Dengan BowlingBall dipilih dalam tampilan Hierarchy, di Inspector, pilih Add Component (Gambar 5.10). Pada menu Add Component, terdapat bilah pencarian. Pada bilah pencarian, ketik Rigid dan pilih komponen Rigidbody (Gambar 5.11).



Gambar 5.9 Bowlingball Yang Diposisikan Ulang Dan Diubah Skalanya

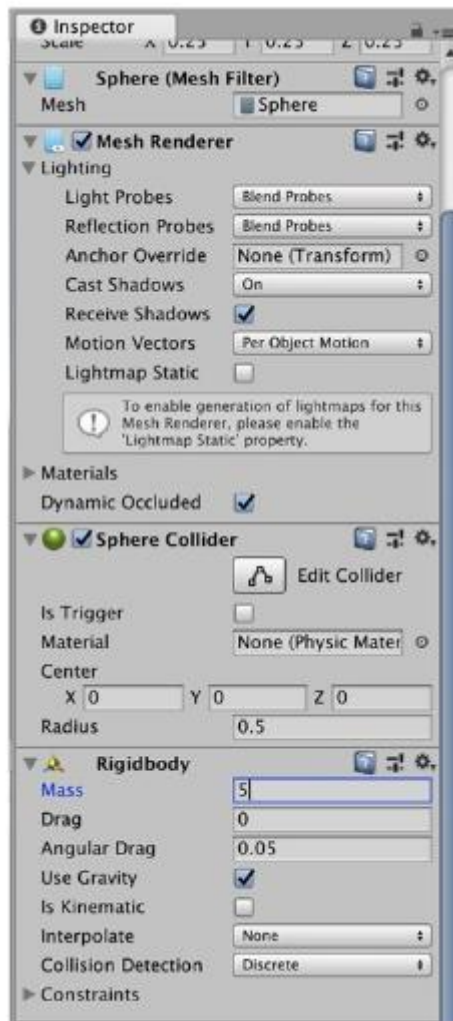


Gambar 5.10 Menambahkan Komponen



Gambar 5.11 Mencari Komponen Rigidbody

Pengaturan standar Komponen Rigidbody cukup bagus, tetapi massa 1 kilogram agak ringan untuk bola bowling. Mari kita atur Massa ke 5, karena 5 kg sudah cukup tepat untuk bola bowling. Pengaturan lainnya sudah bagus untuk saat ini (tetapi kita mungkin harus menyesuainya nanti setelah pengujian). Gambar 5.12 menunjukkan pengaturan untuk BowlingBall.



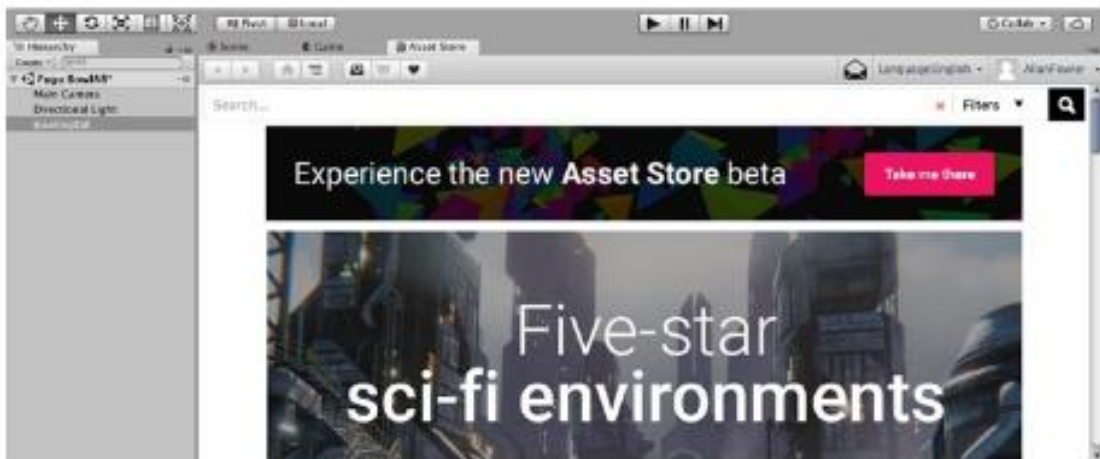
Gambar 5.12 Pengaturan Rigidbody BowlingBall

Pada Gambar 5.12, kita dapat melihat bahwa BowlingBall kita memiliki komponen yang disebut Sphere Collider. Komponen Collider menyediakan GameObject kita dengan bentuk tabrakan. Dalam kasus ini, saat kita membuat GameObject Sphere, collider berbentuk Sphere juga ditambahkan. Komponen Collider akan menentukan pada titik mana GameObject akan bertabrakan. Namun, collider tidak menentukan bagaimana ia akan bertabrakan (memantul, meluncur, atau tidak bergerak sama sekali).

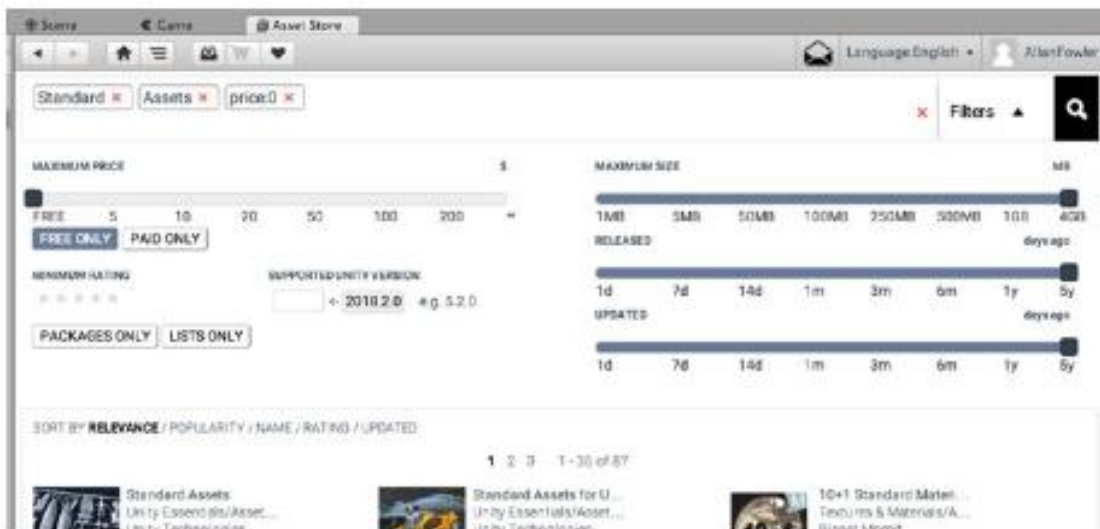
Untuk mengatur bagaimana GameObject bereaksi terhadap Tabrakan, kita perlu menambahkan Material Fisika. Untuk mendapatkan material fisika, kita perlu mengunduh paket Unity Standard Assets dari Unity Store (kita dapat menambahkannya ke Proyek kita saat pertama kali membuatnya). Ada material Fisika lain yang tersedia di Asset Store, beberapa di antaranya gratis, dan yang lainnya harus Anda bayar.

Membuka Asset Store

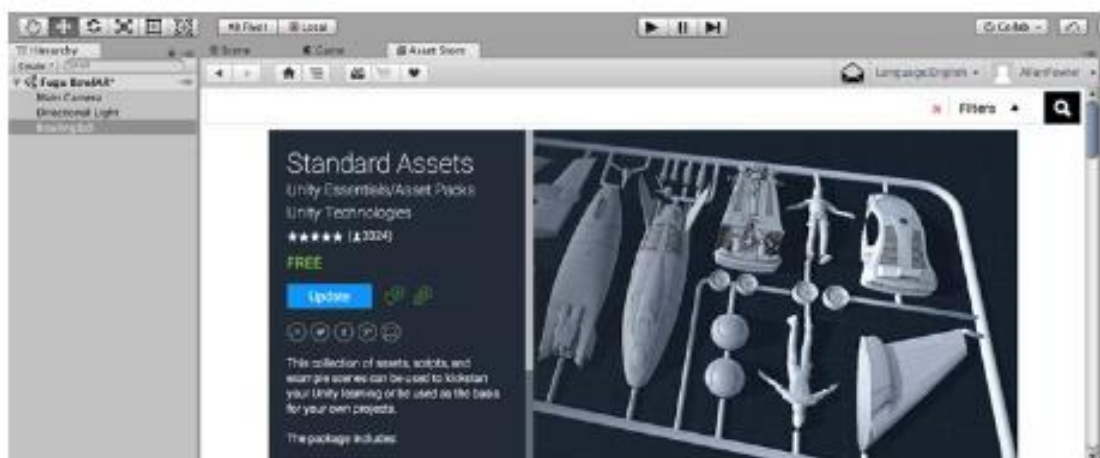
Pilih tab tampilan Asset Store untuk menampilkan Asset Store (Gambar 5-13). Di bilah pencarian, ketik Aset Standar. Jika Anda ingin menambahkan filter untuk hanya menampilkan aset gratis, hal itu akan membatasi hasil pencarian (Gambar 5.14).



Gambar 5.13 Tampilan Unity Asset Store



Gambar 5.14 Mencari Aset Standar (Gratis)



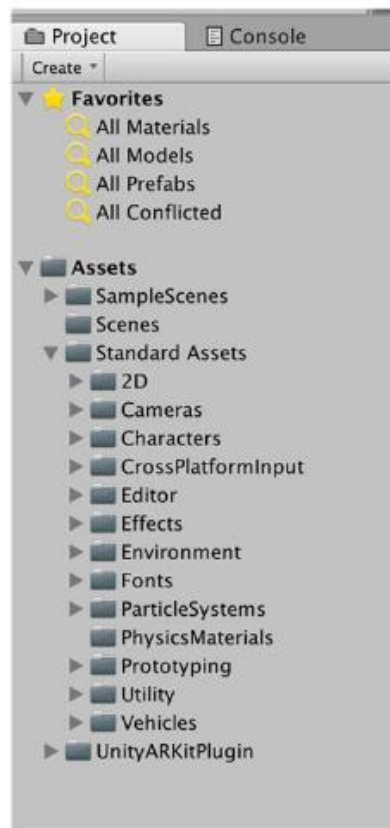
Gambar 5.15 Layar Informasi Paket

Pilih paket Aset Standar dan ini akan membuka layar informasi pada paket ini (Gambar 5.15). Pilih Instal (atau Perbarui jika Anda telah menginstal ini saat membuat Proyek). Setelah beberapa menit (tergantung pada kecepatan unduh Anda), Unity akan membuka Menu Impor paket Unity (Gambar 5.16). Meskipun kita tidak memerlukan semua aset ini, mari kita pilih Impor (kita selalu dapat menghapus apa pun yang tidak kita perlukan atau inginkan nanti).



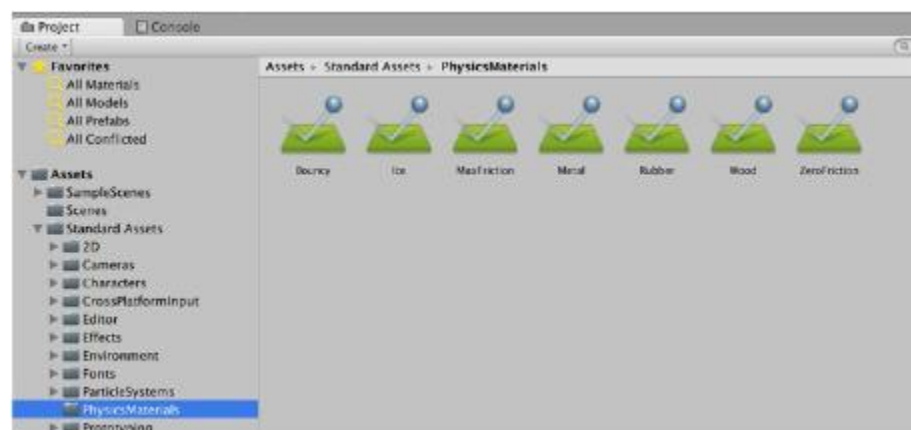
Gambar 5.16 Layar Unity Import Package

Setelah beberapa menit, paket akan diimpor, dan Anda akan melihat beberapa aset tambahan di folder Project (Gambar 5.17).



Gambar 5.17 Paket Aset Standar Unity Dalam Folder Proyek

Jika kita memilih folder `PhysicsMaterials`, kita dapat melihat bahwa sekarang kita memiliki pilihan `PhysicsMaterials` untuk dipilih (Gambar 5.18).

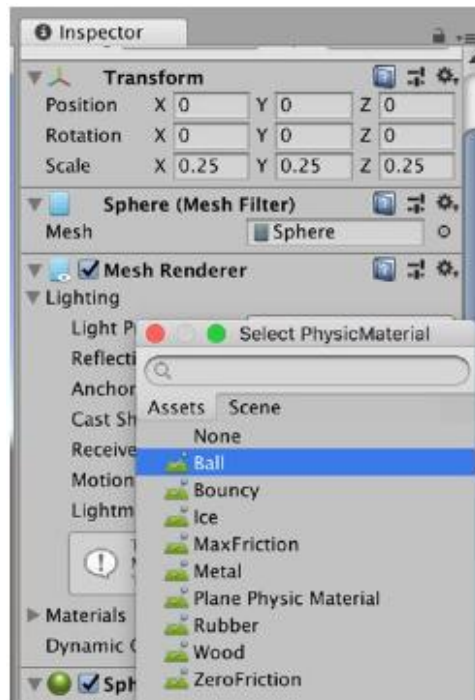


Gambar 5.18 Folder `PhysicsMaterials` dari Unity Standard Assets Package

PhysicsMaterials

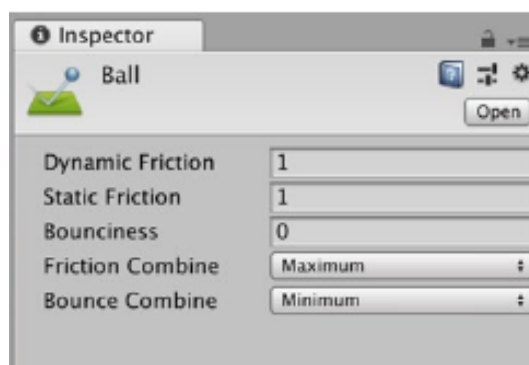
Paket Aset Standar telah memberi kita basis `PhysicsMaterials` untuk digunakan. Akan tetapi, tidak satu pun dari paket tersebut menyediakan `PhysicsMaterial` terbaik untuk BowlingBall kita. Untuk saat ini, mari kita gunakan `PhysicsMaterial` Kayu. Karena kita mungkin perlu menggunakan material ini untuk bagian lain dari permainan, mari kita duplikat. Dari menu utama, pilih Edit ► Duplicate (atau Command+D). Sekarang di folder Project, pilih

PhysicsMaterial duplikat (Wood 1) dan ganti namanya menjadi Ball (Gambar 5.19).



Gambar 5.19 Memilih Ball PhysicsMaterial

Sekarang pilih BowlingBall GameObject dan di Hierarchy, pilih roda gigi kecil di samping kotak properti Material di komponen Sphere Collider lalu pilih Ball PhysicsMaterial yang baru saja kita buat (secara teknis diduplikasi) (Gambar 5.19). Di tab Project klik dua kali Ball PhysicsMaterial untuk mengatur properti fisika. Atur Dynamic Friction ke 1, Static Friction ke 1, Bounciness ke 0, Friction Combine ke Maximum, dan Bounce Combine ke Minimum (Gambar 5.20).

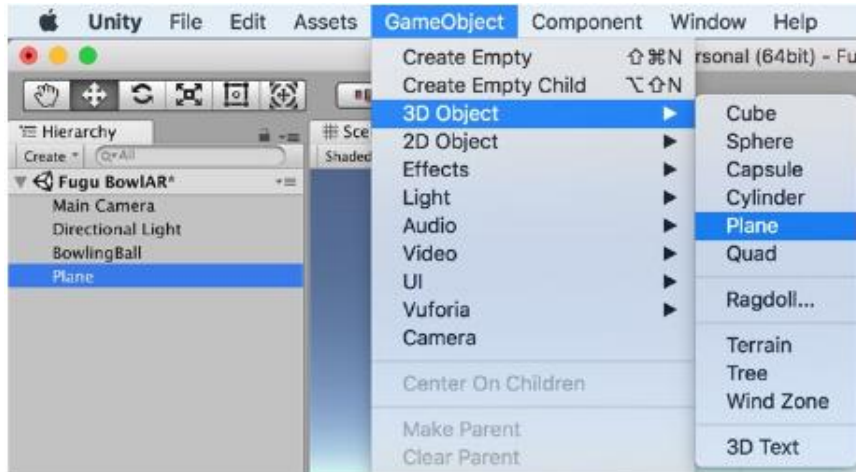


Gambar 5.20 Mengatur Properti PhysicsMaterial Bola

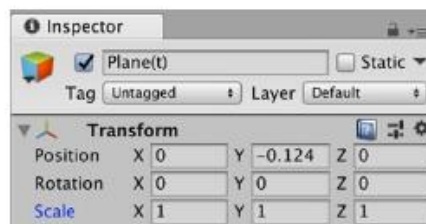
5.3 MEMBUAT BIDANG (SEMENTARA)

Jika kita menguji permainan kita sekarang, kita akan melihat bola kita jatuh (dan tidak akan pernah berhenti). Meskipun, kita akan menggunakan permukaan dunia nyata, untuk saat ini, mari kita pasang Objek Game Bidang sementara untuk membantu kita menguji permainan

kita dalam tampilan Permainan. Dari menu utama, pilih Objek Game ► Objek 3D ► Bidang (Gambar 5.21). Pilih Bidang dalam Hierarki dan ubah namanya menjadi Bidang(t). Jika Anda melihat dalam tampilan Adegan, Anda akan melihat bahwa bidang telah dibuat di Titik Asal dan, sebagai hasilnya, memotong Bola Bowling kita. Mari kita posisikan ulang Bidang(t) kita sehingga berada tepat di bawah Bola Bowling (Gambar 5.22).



Gambar 5.21 Membuat Bidang

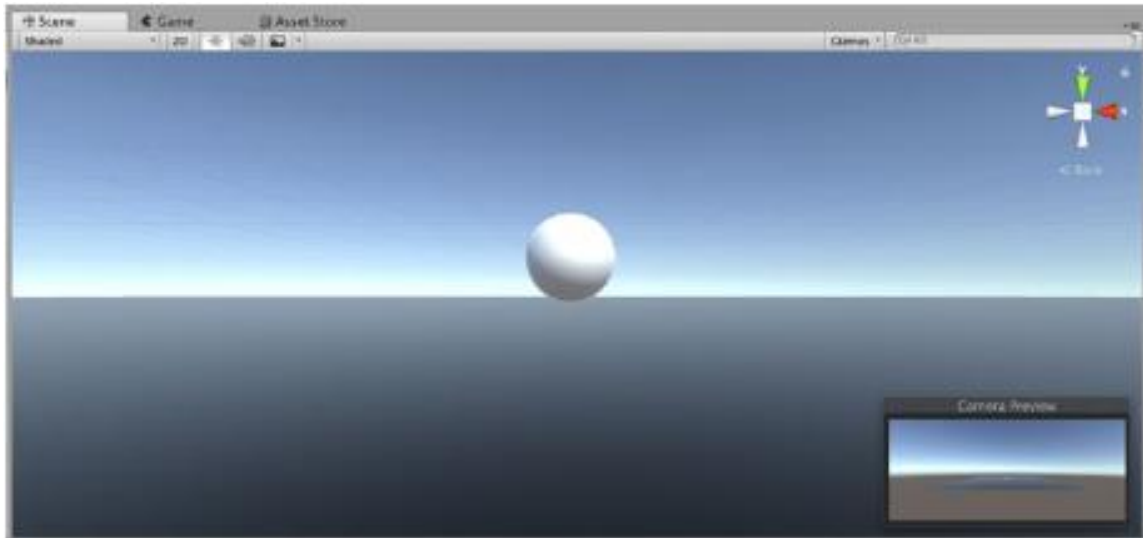


Gambar 5.22 Pengaturan Transformasi Bidang(t)

Jika kita melihat tampilan Adegan, warna Bidang(t) dan Bola Bowling adalah sama. Mari kita ubah warna Bidang(t). Dalam folder Proyek, cari Material (Gambar 5.23), Anda akan melihat ada beberapa aset yang dapat kita gunakan; Saya akan menggunakan BahanBola untuk Bidang(t). Ada beberapa cara untuk mengubah properti Material, tetapi yang termudah adalah dengan memilih dan menyeret serta meletakkan BahanBola ke Bidang(t) dalam tampilan Adegan (Gambar 5.24).

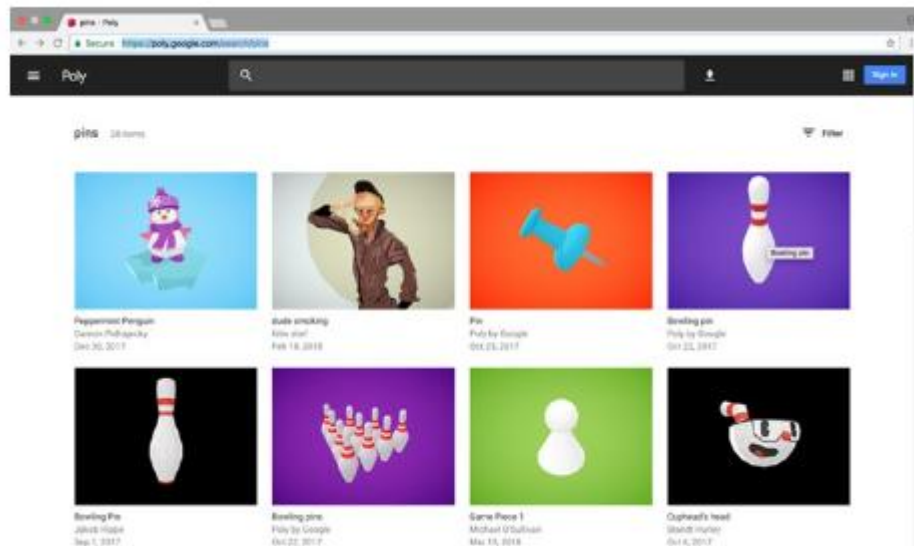


Gambar 5.23 Mencari Ballmaterial



Gambar 5.24 Bidang(T) Dengan Ballmaterial

Sekarang, mari tambahkan beberapa Pin (atau Skittles tergantung dari mana Anda berasal). Jika kita mencari Pin Bowling di Asset Store, kita akan menemukan bahwa satu-satunya pilihan adalah Aset Berbayar. Namun, teman baik kita di Google memiliki pustaka aset AR dan VR yang fantastis di <https://poly.google.com/>. Buka situs ini dengan browser favorit Anda dan cari Pin. Anda akan menemukan beberapa pilihan yang bagus (Gambar 5.25).

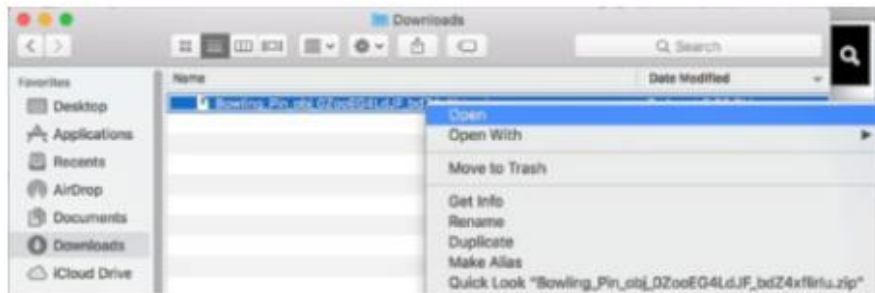


Gambar 5.25 Mencari Pin di Google Poly

Pilih Pin yang ingin Anda gunakan; Saya akan menggunakan Bowling Pin dari Poly by Google. Jika kita memeriksa lisensinya, kita dapat melihat bahwa pin tersebut tersedia di bawah Lisensi Creative Commons, yang memungkinkan kita untuk menggunakannya (lihat <https://creativecommons.org/> untuk detail lebih lanjut). Pilih tautan Unduh dan pilih OBJ untuk mengunduh gambar ini. Saat Anda berada di halaman Google Poly, lihat beberapa konten yang menakutkan.

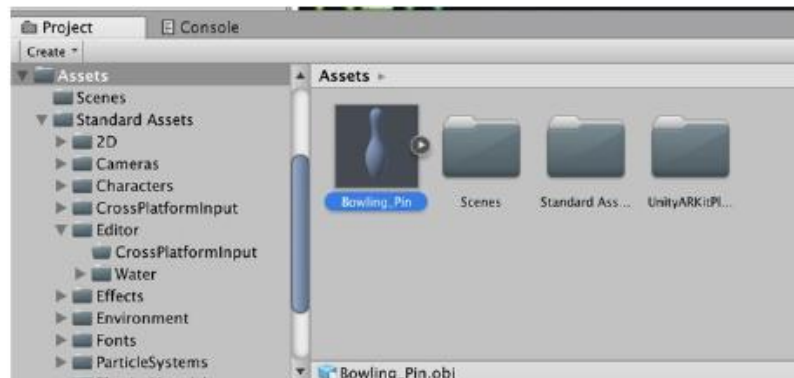
5.4 MENGIMPOR FILE OBJ KE UNITY

Setelah file selesai diunduh, buka folder tempat file tersebut disimpan dan klik kanan (atau klik dua kali) untuk mengekstrak folder tersebut (Gambar 5.26). Di folder tersebut, Anda akan melihat bahwa ada file OBJ (yang kita butuhkan). Sekarang di Unity, pilih dari menu utama Aset ► Impor Aset Baru dan telusuri folder yang telah diekstrak dengan file Pin OBJ dan pilih file OBJ dan pilih Impor.



Gambar 5.26. Mengekstrak folder Bowling_Pin.obj

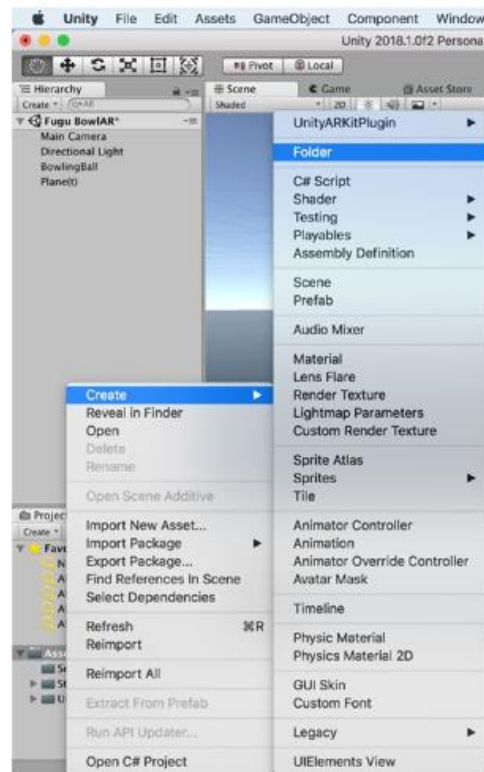
Setelah Unity selesai mengimpor aset ini, aset tersebut akan muncul di folder Aset (Gambar 5.27). Saya akan membuat folder baru untuk menyimpan Art yang diimpor yang mungkin saya gunakan. Jika Anda tidak menemukan aset ini di folder Aset, pastikan untuk mencarinya di tab Tampilan Proyek.



Gambar 5.27 Aset Bowling_Pin di Folder Aset

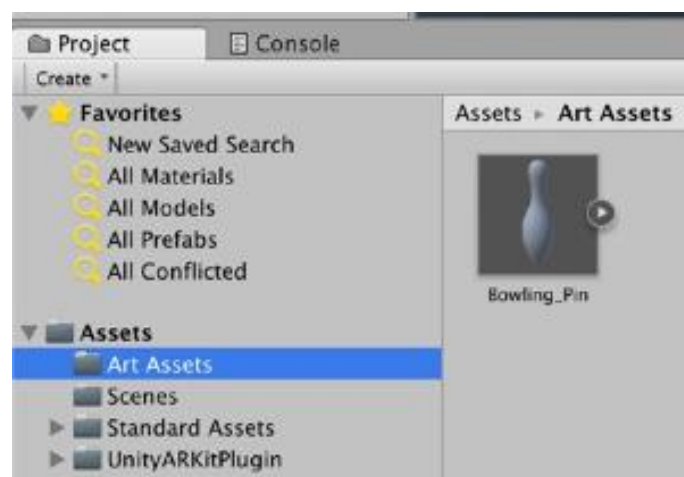
Membuat Folder Proyek Baru

Untuk membuat folder baru di tampilan Proyek, di Folder Proyek klik kanan dan pilih dari menu dan pilih Buat ► Folder (Gambar 5.28).



Gambar 5.28 Membuat Folder Proyek Baru

Unity akan membuat folder di folder Proyek Aset, dan sekarang kita perlu menamainya dan menyeret serta meletakkan aset `Bowling_Pin` ke folder ini. Saya akan menamai folder ini Aset Seni (Gambar 5.29).



Gambar 5.29 Folder Aset Seni dengan aset `Bowling_Pin`

Menambahkan `Bowling_Pin` ke Adegan

Dengan aset `Bowling_Pin` dipilih, seret ke Hierarki (atau tampilan Adegan) (Gambar 5.30). Anda akan melihat bahwa sekali lagi, kita perlu Mengubah posisi dan skala `GameObject` ini.

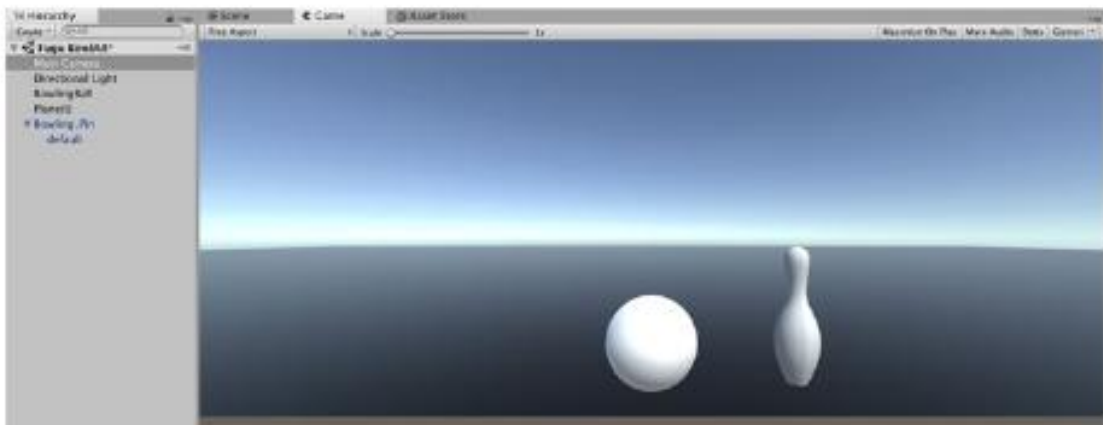


Gambar 5.30 Bowling_Pin dalam Hierarki

Pengaturan Transformasi untuk Bowling_Pin dan BowlingBall

Pertama-tama, mari kita skalakan Bowling_Pin kita, karena saat ini ditetapkan pada skala 1,1,1. Namun, dimensi yang benar adalah lebar 121 mm (pada titik terlebar) dan tinggi 380 mm. Mari kita skalakan Bowling_Pin ke 0,4,0,4,0,4. Skalanya tidak sempurna, tetapi terlihat cukup bagus. Mari kita juga skalakan BowlingBall ke skala relatif.

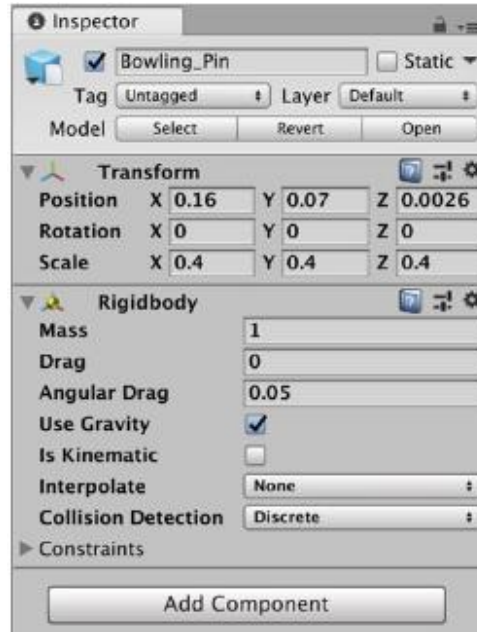
Menurut Wikipedia, Bola bowling mungkin memiliki keliling antara 67,83 cm dan 68,59 cm, dan diameter dalam kisaran 21,59 cm hingga 21,83 cm. Mari kita atur skala BowlingBall kita ke 0,1,0,1,0,1. Skalanya tidak sepenuhnya sesuai, tetapi cukup bagus untuk saat ini. Gambar 5-31 menunjukkan BowlingBall saya di samping BowlingPin saya. Keduanya terlihat cukup bagus (bagi saya).



Gambar 5.31 Tampilan Game dari BowlingBall dan Bowling_Pin yang telah diubah skalanya

Menambahkan Rigidbody ke Bowling_Pin

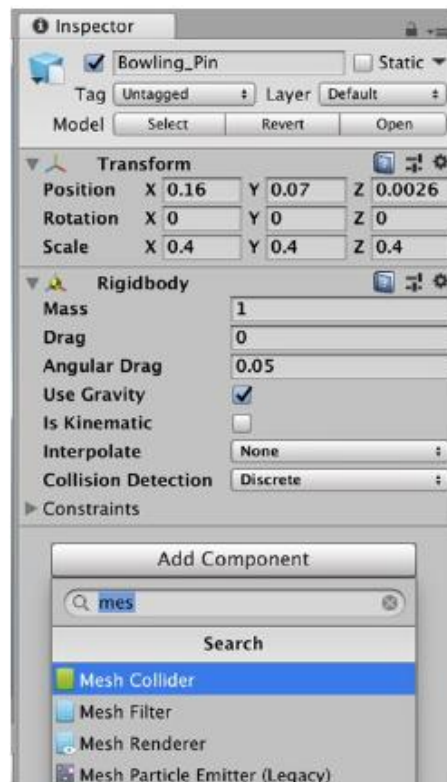
Kita akan mengikuti proses yang sama untuk menambahkan Rigidbody dan PhysicsMaterial ke Bowling_Pin seperti yang kita lakukan dengan BowlingBall. Dengan Bowling_Pin yang dipilih dalam Hierarchy, di Inspector, pilih Add Component dan cari serta tambahkan komponen Rigidbody (Gambar 5.32).



Gambar 5.32 GameObject Bowling_Pin dengan komponen Rigidbody

Menambahkan Collider ke Bowling_Pin

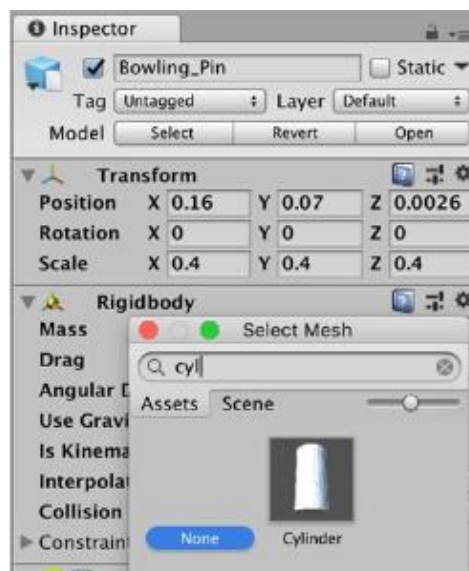
Jika kita melihat Inspector Bowling_Pin kita, Anda akan melihat bahwa tidak ada Collider. Itu karena Bowling_Pin tidak disertai Collider karena bentuknya tidak cembung. Kita akan Menambahkan Collider, sehingga kita dapat menambahkan PhysicsMaterial. Dengan Bowling_Pin yang dipilih, di Inspector, pilih Add Component dan cari mesh (Gambar 5.33).



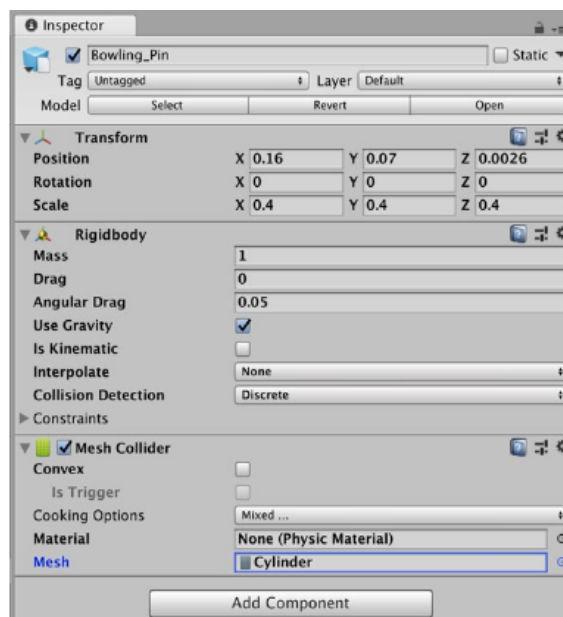
Gambar 5.33 Mencari komponen Mesh Collider

Sekarang, pilih dan tambahkan komponen Mesh Collider. Mesh Collider akan memungkinkan kita membuat mesh yang akan mengelilingi Game Object. Dalam komponen Mesh Collider yang baru saja kita tambahkan, terdapat kotak properti untuk kita mengatur jenis mesh yang ingin kita gunakan. Komponen mesh collider memungkinkan kita menambahkan mesh yang berbentuk cembung (silinder, kubus, bola, atau bidang).

Karena Bowling_Pin kita paling mirip dengan bentuk silinder, mari kita cari jaring silinder dan tambahkan ini ke GameObject Bowling_Pin. Pilih roda gigi kecil di sebelah kanan kotak properti pengaturan Mesh dari Mesh Collider dan cari Cylinder (Gambar 5.34). Pilih jaring Cylinder untuk menetapkan ini sebagai bentuk Mesh Collider (Gambar 5.35). Perhatikan juga, saya telah menetapkan komponen Rigidbody sebagai Kinematic.



Gambar 5.34 Mencari Jaring Silinder



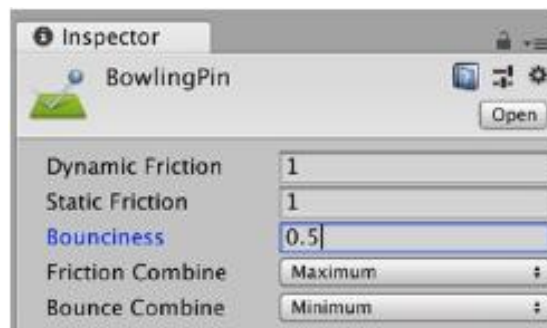
Gambar 5.35 Set Jaring Silinder sebagai Penumbuk Jaring

Menambahkan PhysicsMaterial ke Bowling_Pin

Sama seperti yang kita lakukan untuk BowlingBall, kita perlu menambahkan properti physics untuk mengatur bagaimana pin bowling kita bereaksi terhadap tumbukan. Karena properti physics dari pin bowling mirip dengan bola bowling, mari kita gandakan PhysicsMaterial Bola dan ganti namanya menjadi BowlingPin dan ubah pengaturannya. Pertama, pilih dan gandakan PhysicsMaterial bola dan beri nama BowlingPin (Gambar 5.36). Pilih PhysicsMaterial BowlingPin dan ubah pengaturannya menjadi berikut; Dynamic Friction = 1, Static Friction = 1, Bounciness = 0.5, Friction Combine = Maximum, Bounce Combine = Minimum (Gambar 5.37).

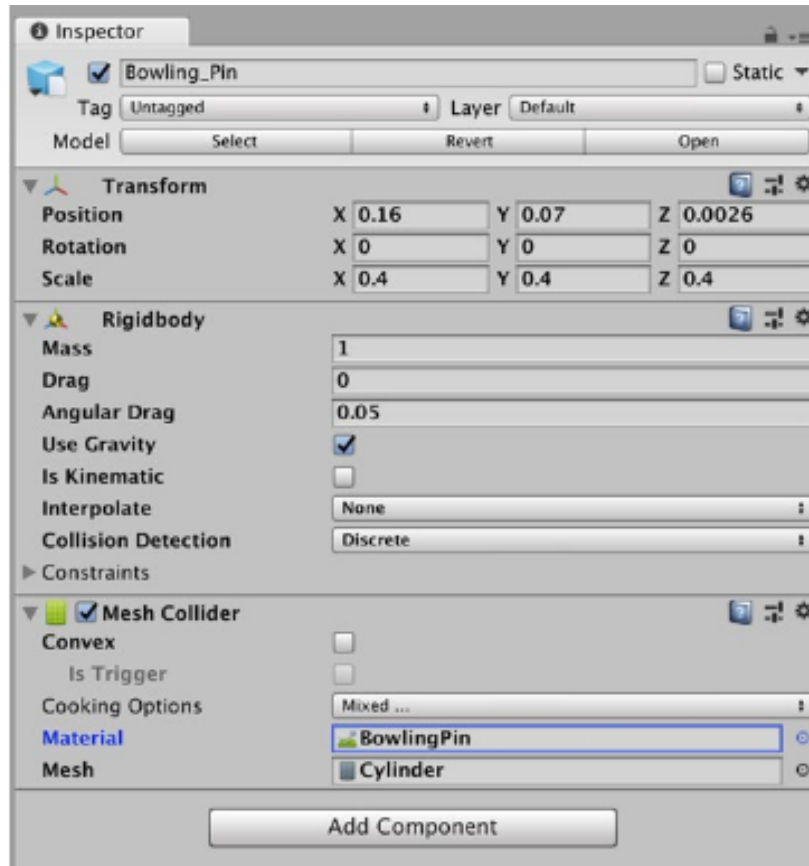


Gambar 5.36 Bowlingpin Physicsmaterial



Gambar 5.37 Pengaturan Bowlingpin Physicsmaterial

Sekarang tambahkan BowlingPin PhysicsMaterial ke BowlingPin GameObject Mesh Collider (Gambar 5.38).



Gambar 5.38 Set BowlingPin PhysicsMaterial sebagai Material

Membuat Bola Bowling Bergulir

Untuk membuat Bola Bowling kita bergulir berdasarkan masukan Pemain, kita perlu membuat skrip. Seperti yang telah saya sebutkan, ini bukanlah buku Pemrograman Pengantar, jadi saya tidak akan menghabiskan banyak waktu untuk membahas tujuan setiap baris kode. Mungkin ada beberapa bagian penting dari kode yang menurut saya perlu dicatat, dan saya akan menyertakan komentar dalam Skrip untuk menjelaskan tujuan dari baris kode tersebut. Pertama-tama mari kita buat folder untuk menyimpan semua Skrip kita. Dalam tampilan Proyek, klik kanan dan pilih Buat ► Folder.

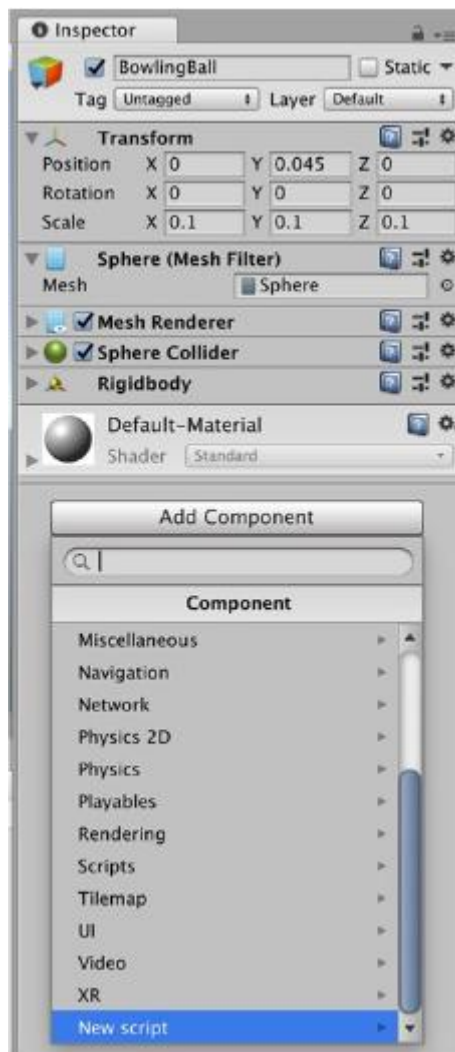
Mari kita beri nama Skrip (Gambar 5.39). Anda mungkin ingin memberi nama folder ini dengan sesuatu yang benar-benar lucu atau keren, tetapi sebaiknya gunakan konvensi penamaan file sehingga jika orang lain perlu mengedit atau memodifikasi permainan Anda, mereka dapat dengan mudah mengidentifikasi tempat menemukan aset dalam proyek Anda. Untuk saat ini, kita akan membuat skrip yang memungkinkan kita menguji masukan pengguna di Unity. Hasilnya, kita akan membuat GameObject BowlingBall kita bergerak berdasarkan masukan Keyboard (atau Mouse).

Kita akan mengubahnya nanti sehingga masukannya berdasarkan Sentuhan, Ketukan, atau Gesek. Di Hierarki, pilih GameObject BowlingBall kita dan pilih Tambahkan Komponen. Di Tambahkan Komponen, pilih Skrip Baru (Gambar 5.40). Pertama, beri nama Skrip ini PlayerController dan pilih Buat dan Tambahkan. Opsi ini memungkinkan kita membuat Skrip dan Menambahkannya ke GameObject BowlingBall kita. Unity akan menempatkan Skrip ini di

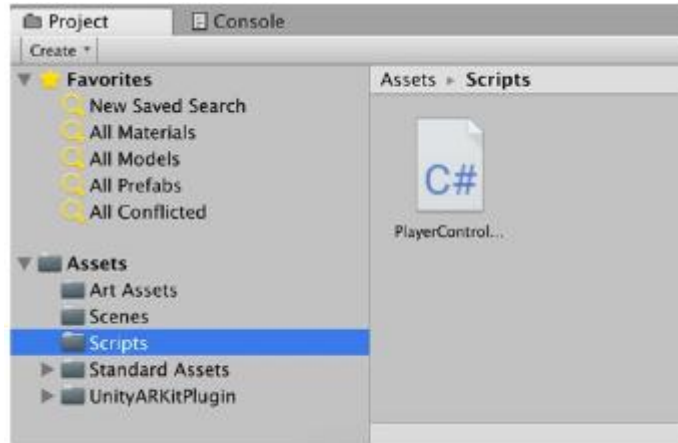
folder Aset. Kita perlu memilih dan memindahkan (menyeret dan melepas) skrip ini ke menu Skrip kita (Gambar 5.41).



Gambar 5.39 Folder Skrip



Gambar 5.40 Menambahkan Komponen Skrip Baru



Gambar 5.41 Skrip PlayerController yang ditambahkan ke folder Skrip

5.5 MENEGIT SKRIP PLAYERCONTROLLER

Sekarang mari kita edit Skrip PlayerController. Jika Anda memilih Skrip PlayerController, Anda dapat melihat kode untuk aset ini di Inspektur. Mari kita edit kode ini. Klik dua kali skrip PlayerController dan ini akan membuka IDE default Anda. Saya menggunakan Visual Studio. Masukkan kode berikut seperti yang ditunjukkan pada Daftar 5.1.

Daftar 5.1 Skrip PlayerController

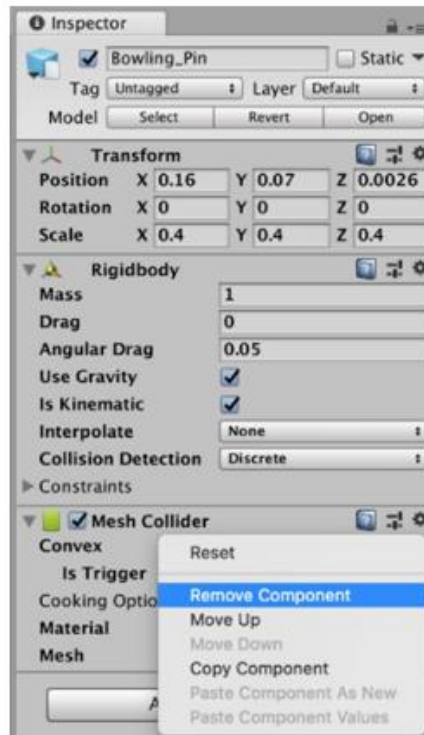
```
using UnityEngine;
using System.Collections;
public class PlayerController : MonoBehaviour {
public float speed;
private Rigidbody rb;
void Start ()
{
rb = GetComponent<Rigidbody>();
void FixedUpdate ()
{
}
float moveHorizontal = Input.GetAxis ("Horizontal");
float moveVertical = Input.GetAxis ("Vertical");
Vector3 movement = new Vector3 (moveHorizontal, 0.0f,
moveVertical);
rb.AddForce (movement * speed);
}
}
```

Sekarang kita siap untuk melihat apakah kode kita berfungsi. Jika kita menjalankan dan menguji permainan kita saat kita menekan salah satu tombol panah, BowlingBall bergerak ke kedua arah (yang bagus, ya, karena kita hanya perlu bola bergerak ke satu arah). Namun, Anda akan melihat bahwa collider pada Bowling_Pin terlalu besar, dan tabrakannya tidak realistis.

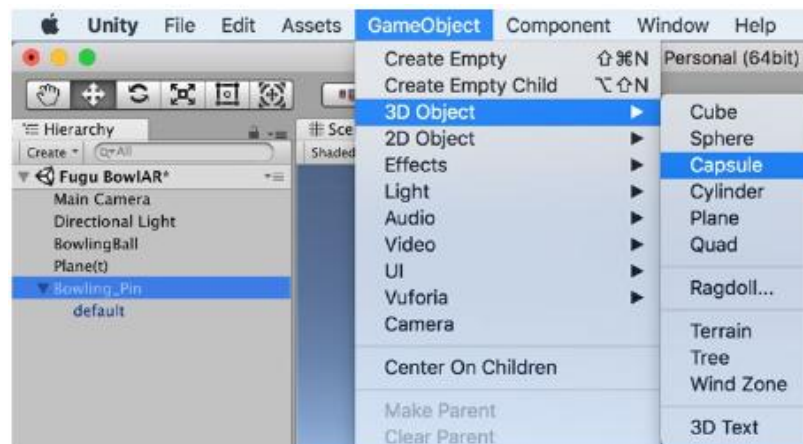
Sekarang, kita dapat mengedit collider ini, tetapi kita akan membuat collider kita sendiri.

Membuat Collider Kita Sendiri

Untuk membuat Collider kita sendiri, kita akan menggunakan sedikit keajaiban desain permainan (juga dikenal sebagai faking it). Untuk melakukannya, di Hierarchy pilih GameObject Bowling_Pin dan di Mesh Collider Component pilih roda gigi kecil dan klik kanan dan hapus Collider saat ini (Gambar 5.42). Sekarang dari menu Utama, pilih GameObject ► 3D Object ► Capsule (Gambar 5.43).



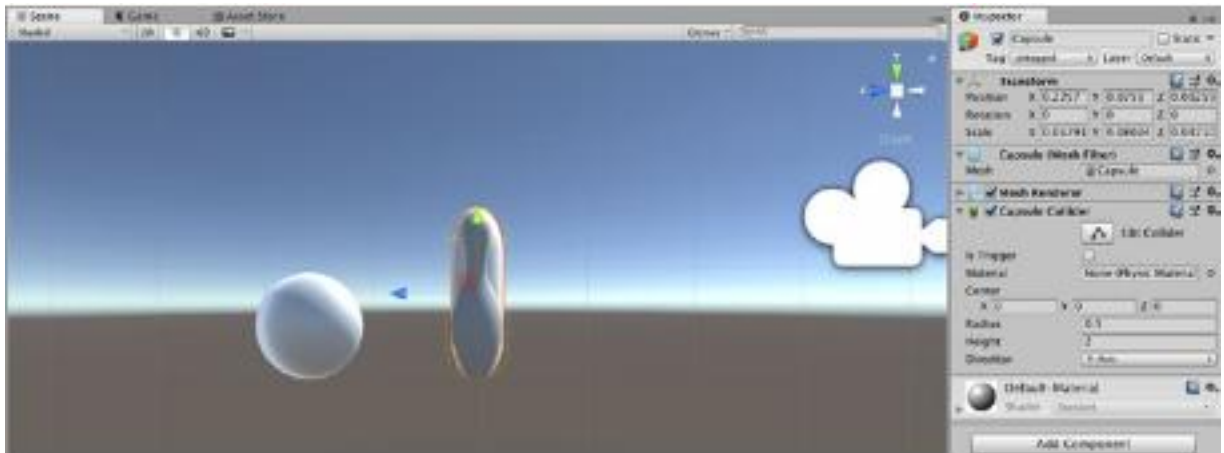
Gambar 5.42 Skrip Playercontroller Ditambahkan Ke Folder Skrip



Gambar 5.43 Membuat Gameobject Kapsul 3D

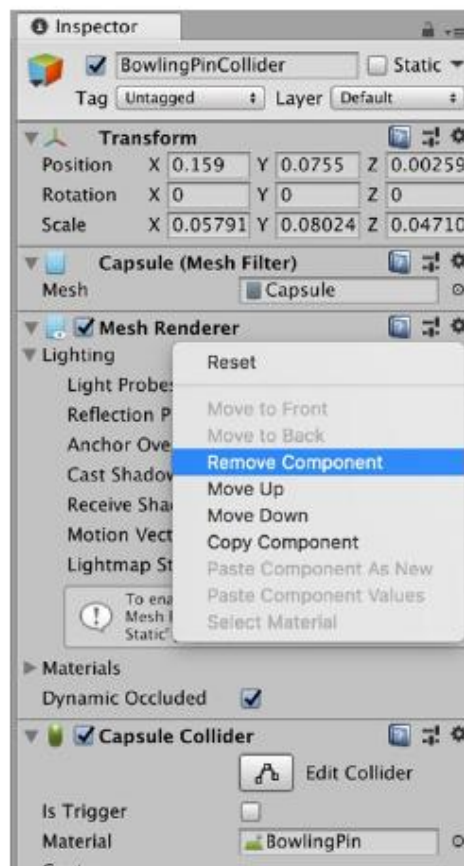
Anda akan melihat bahwa GameObject Kapsul kita agak besar dan oleh karena itu, kita perlu Mengubah Skala. Perhatikan juga, bahwa Kapsul memiliki Collider-nya sendiri. Pada Gambar

5.44, saya telah Mengubah Skala dan Posisi, sehingga tinggi dan lebarnya hampir sama dengan Bowling_Pin.



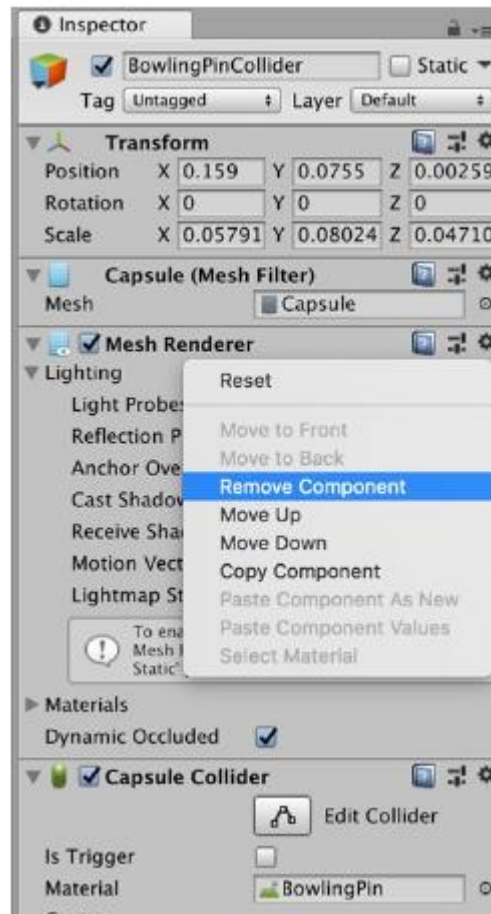
Gambar 5.44 Kapsul yang Diubah Skala dan Posisinya

Sekarang kita tambahkan PhysicsMaterial ke Capsule kita, sehingga dapat merespons gaya BowlingBall kita. Untuk saat ini, mari kita gunakan PhysicsMaterial BowlingPin. Dengan Capsule yang dipilih, di Inspector, pilih roda gigi kecil di sebelah kotak Material properties dan klik kanan dan tambahkan PhysicsMaterial BowlingPin (Gambar 5.45).



Gambar 5.45 Menambahkan Material Fisika BowlingPin

Dengan GameObject Capsule yang dipilih, mari beri nama yang bermakna. Saya akan menggunakan BowlingPinCollider. Sekarang setelah kita memindahkan BowlingPinCollider sehingga menutupi GameObject Bowling_Pin secara menyeluruh, langkah selanjutnya adalah membuatnya tidak terlihat oleh pemain. Untuk membuat GameObject BowlingPinCollider tidak terlihat oleh pemain, di Hierarchy pilih GameObject BowlingPinCollider dan di Inspector, di Mesh Renderer Component, pilih roda gigi di sebelah kanan dan pilih Remove Component (Gambar 5.46).



Gambar 5.46 Menghapus Komponen Mesh Renderer

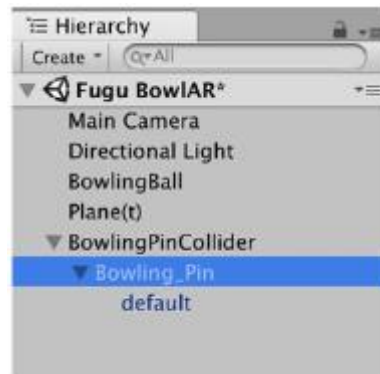
Mengingat Parent

Anda mungkin ingat sebelumnya di Bab ini, saya membahas Parent dan Children dalam pengembangan perangkat lunak. Kita akan menjadikan Bowling_Pin sebagai Child dari BowlingPinCollider. Dengan cara ini, apa pun yang terjadi pada BowlingPinCollider juga akan terjadi pada Bowling_Pin.

Membuat Parent

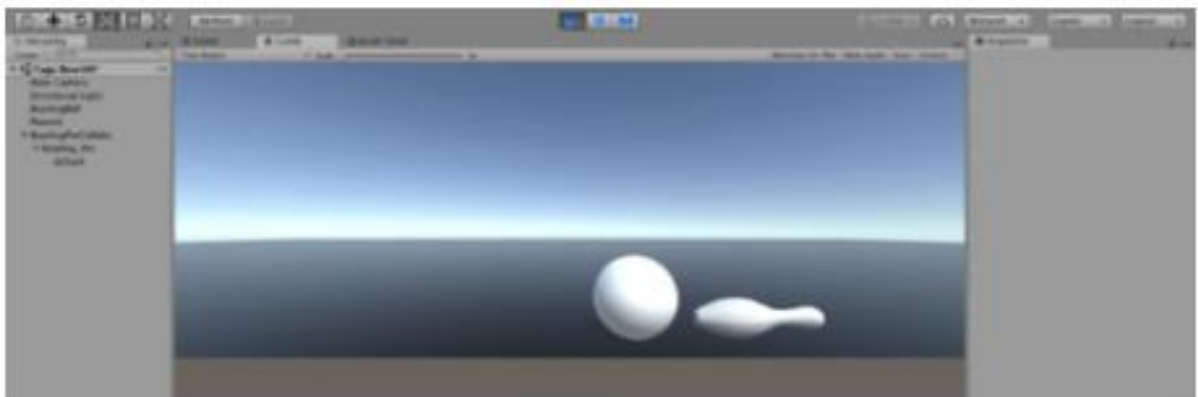
Menjadi parent adalah salah satu pengalaman paling seru dan menakutkan dalam hidup saya. Untungnya, Unity membuat proses pembuatan Parent GameObject jauh lebih mudah (dan mungkin juga jauh lebih mudah). Untuk menjadikan BowlingPinCollider sebagai Parent dari Bowling_Pin GameObject, kita cukup menyeret Bowling_Pin GameObject ke BowlingPinCollider dan BowlingPinCollider akan menjadi child dari BowlingPinCollider

(Gambar 5.47). Andai saja hidup semudah itu.



Gambar 5.47 BowlingPinCollider sebagai Induk dari Bowling_Pin

Sekarang mari kita uji pekerjaan kita untuk melihat apakah Collider kita berfungsi sebagaimana mestinya. Wah, collider saya berfungsi (Gambar 5.48). Namun, BowlingPinCollider Parent dan Bowling_Pin Child masih belum jatuh. Jujur saja, ini cukup mencerminkan kemampuan bowling saya di dunia nyata. Namun, saya rasa para pemain akan sedikit frustrasi jika mereka tidak dapat menjatuhkan pin bowling. Jadi, mari kita buat beberapa perubahan untuk membuatnya sedikit lebih realistis.

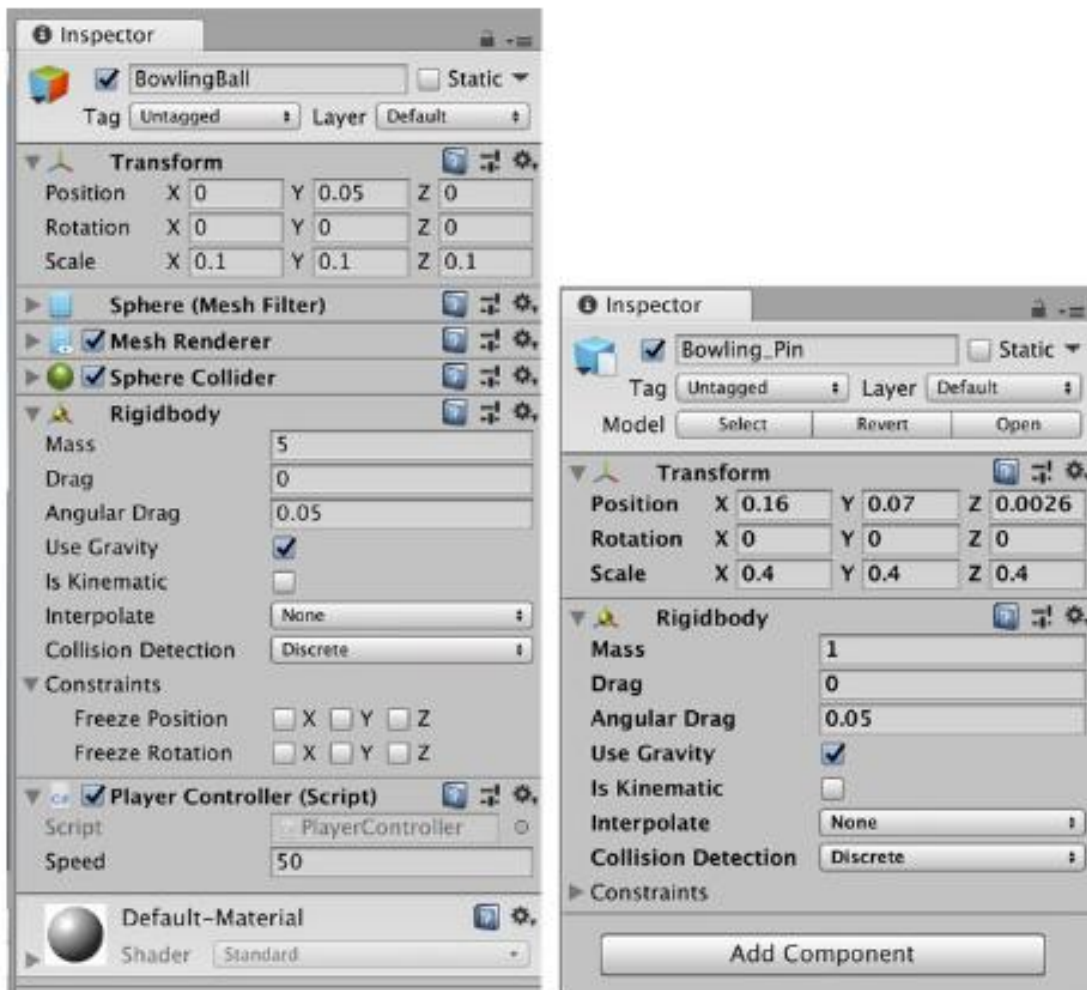


Gambar 5.48 Tampilan Game dari tabrakan

Membuat Bowling_Pin Jatuh

Jika Bowling_Pin Anda tidak jatuh, periksa Komponen Rigidbody pada setiap GameObject. Dalam pengaturan saya, saya telah menetapkan Massa Komponen Rigidbody dari BowlingBall menjadi 5 (artinya BowlingBall berbobot 5 kg). Saya telah menetapkan Bowling_Pin menjadi 1 kg. Secara teknis, Bowling Pin harus berbobot 1,6 kg.

Jadi, saya akan membiarkan pengaturan saya apa adanya. Jika BowlingBall Anda tidak membuat Bowling_Pin jatuh, coba sesuaikan beratnya. Pemain tidak akan pernah mengangkat salah satu dari keduanya secara fisik dan tidak akan pernah tahu berat yang tepat. Ditambah lagi, dalam Desain Game, terkadang menyenangkan untuk terlalu menekankan realitas.



Gambar 5.49 Komponen RigidBody BowlingBall dan Bowling_Pin

Ringkasan

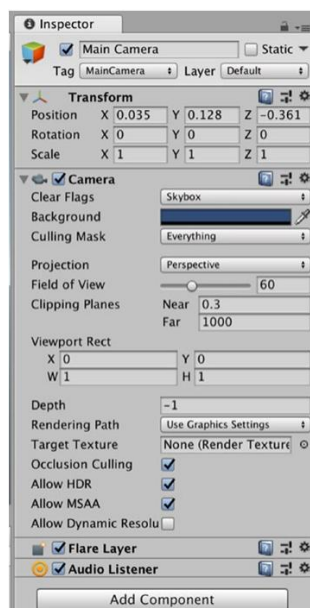
Dalam bab ini, kita telah membuat adegan baru dan menambahkan GameObjects. Kita telah menambahkan Komponen RigidBody dan PhysicsMaterials. Kita telah berhasil menguji permainan kita untuk melihat apakah permainan tersebut berfungsi di GameView. Dalam bab berikutnya, kita akan menguji permainan dalam AR untuk melihat tampilannya.

BAB 6

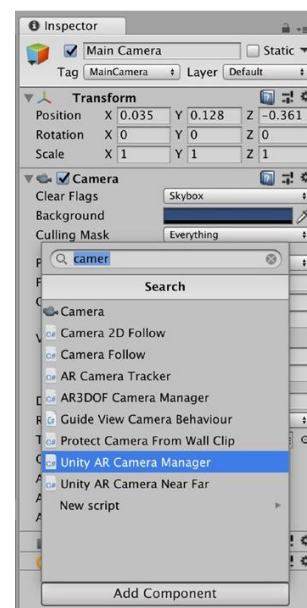
MEMPERKENALKAN SENTUHAN

6.1 MENGUJI GAME DALAM AR

Pada bab terakhir, kita dapat menguji Game kita dalam Unity Game View; sekarang, mari kita lihat seperti apa tampilan game tersebut dalam AR. Dengan Game FuguBowlAR yang terbuka di Unity, mari kita tambahkan Unity AR Camera Manager ke Kamera Utama kita. Di Hierarchy, pilih Kamera Utama, lalu di Inspector, pilih Add Component (Gambar 6.1). Di bilah pencarian Add Component, cari Camera dan pilih Unity AR Camera Manager (Gambar 6.2).



Gambar 6-1. Menambahkan Komponen

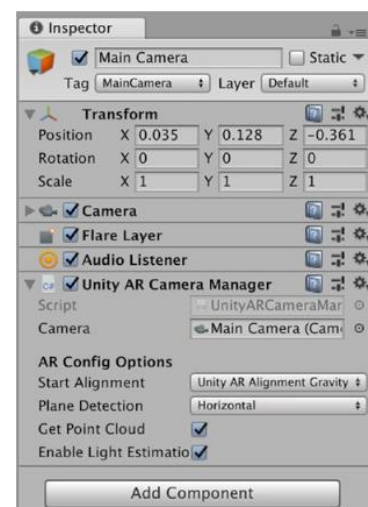


Gambar 6-2. Menambahkan Komponen Unity AR Camera Manager

Setelah Unity AR, Komponen Camera Manager ditambahkan, sekarang pilih dan seret Kamera Utama dari Hierarki dan tambahkan ini ke kotak properti Kamera dari komponen Unity AR Camera Manager (Gambar 6.3).

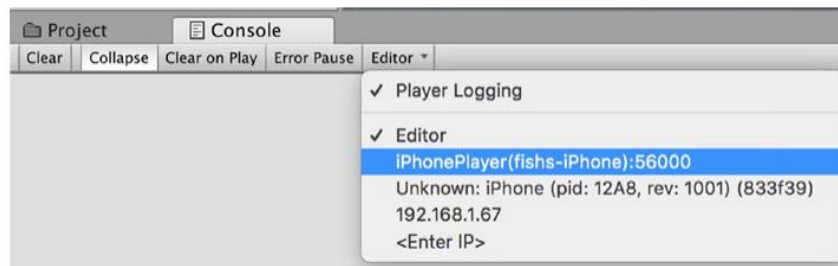
Sekarang setelah komponen Unity AR Camera Manager ditambahkan ke Kamera Utama, mari kita lihat pratinjau seperti apa tampilan game dalam tampilan Game, menggunakan Aplikasi Unity AR Kit Remote yang telah kita instal di perangkat kita. Pertama, hubungkan perangkat Anda ke Mac, lalu jalankan Aplikasi Unity AR Kit Remote di perangkat Anda. Di Unity, pilih tampilan Konsol dan pilih Editor, lalu pilih perangkat Anda (Gambar 6.4).

Sekarang, pilih tombol Play dari tampilan Game Kamera Utama ke Properti Kamera

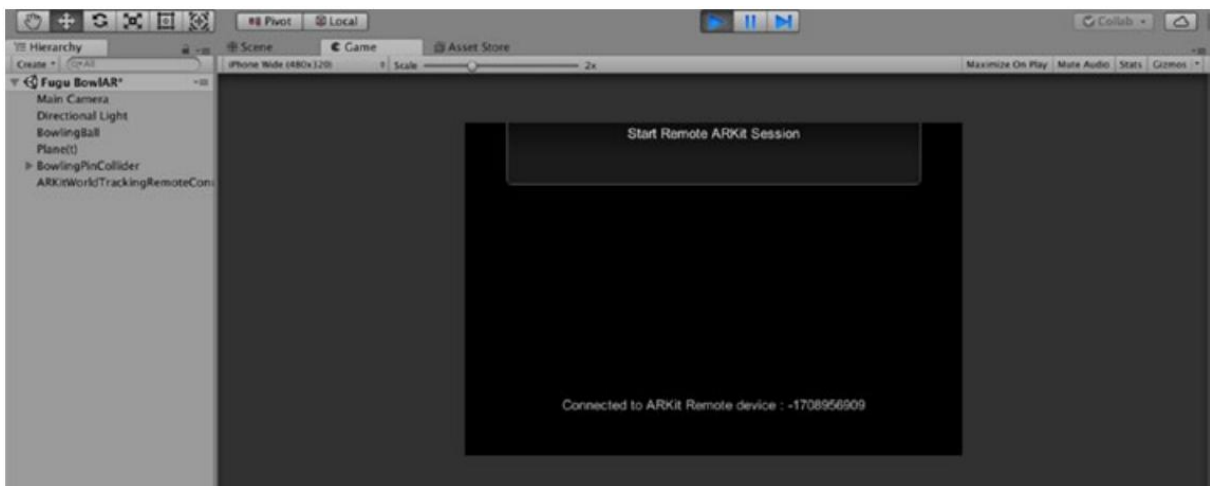


Gambar 6.3. Menambahkan

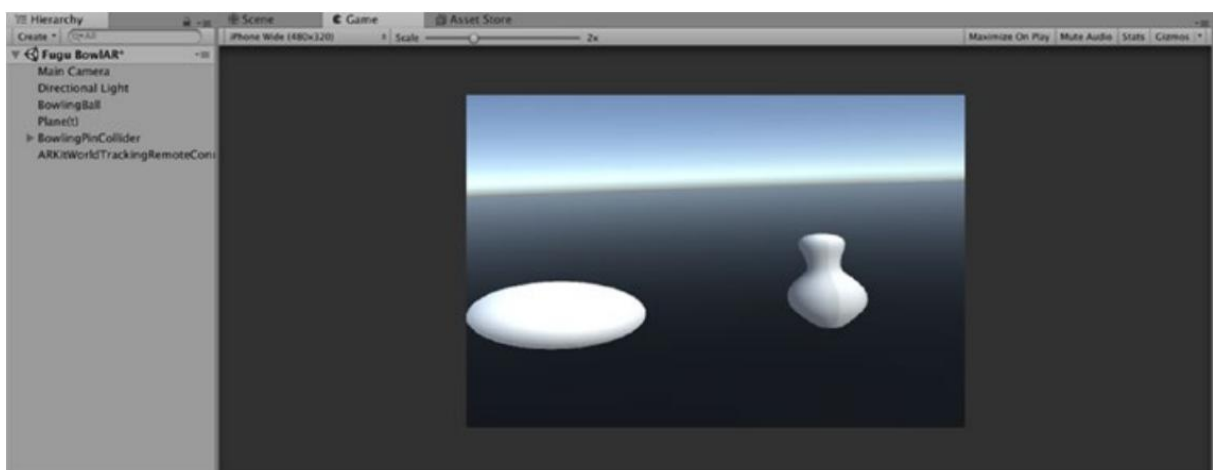
Unity, lalu di tampilan game, pilih (Gambar 6-5). Anda akan melihat game tersebut sekarang berfungsi di tampilan Game Unity (Gambar 6-6).



Gambar 6.4. Menghubungkan ke iPhonePlayer



Gambar 6.5. Memulai Sesi ARKit Jarak Jauh



Gambar 6.6. Tampilan game versi AR

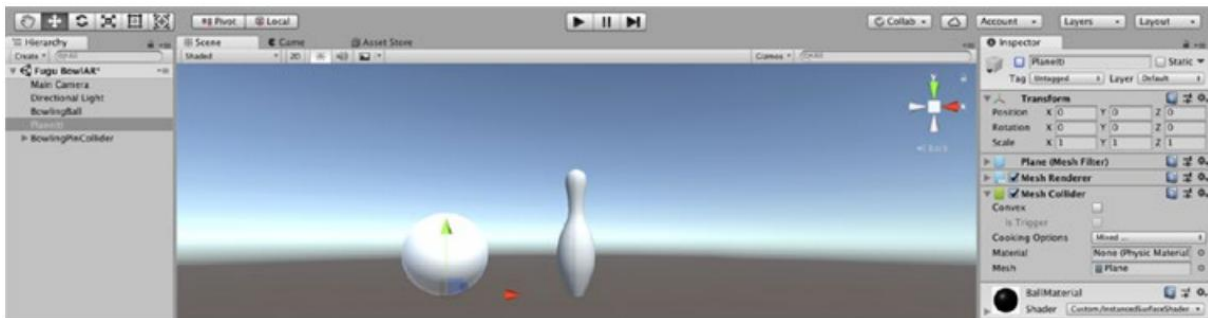
6.2 Beberapa Tantangan

Anda akan melihat ada beberapa tantangan dengan permainan kita. Pertama, seperti yang diharapkan, Bola Bowling tidak bergerak. Ini karena kita awalnya menyetel kode Bola Bowling untuk merespons masukan keyboard. Kita melakukan ini dengan sengaja karena tampilan Permainan Unity (saat ini) tidak memungkinkan kita menguji iterasi pengguna

dengan sentuhan. Tantangan lainnya adalah bahwa Plane(t) kita masih terlihat. Pertama-tama mari kita nonaktifkan Plane(t). Saya tidak ingin menghapusnya sekarang, karena saya mungkin ingin menggunakannya untuk pengujian nanti.

Menonaktifkan Plane(t)

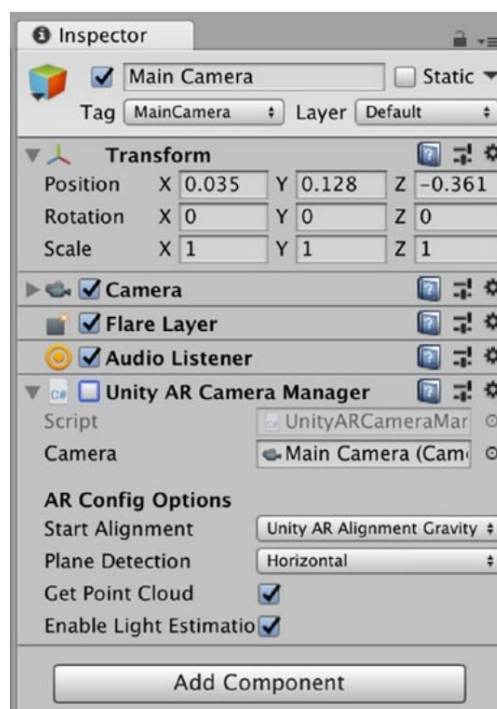
Menonaktifkan GameObject Plane(t) relatif mudah. Dalam tampilan Hierarchy, pilih GameObject Plane(t) lalu di Inspector, pilih kotak centang di sebelah kiri nama GameObject (Gambar 6-7). Anda akan segera melihat bahwa GameObject Plane(t) tidak lagi terlihat. Dampaknya akan terlihat jelas saat kita menguji permainan kita.



Gambar 6-7. GameObject Plane(t) dinonaktifkan

Pengujian

Untuk menguji game kita, ada dua cara. Cara pertama dan paling efisien adalah mengujinya dalam tampilan Game. Untuk melakukannya, kita perlu menonaktifkan Komponen Unity AR Camera Manager yang kita tambahkan ke kamera. Pilih Kamera Utama, lalu hapus centang pada kotak di sebelah kiri nama file (Gambar 6-8).



Gambar 6-8. Unity Camera Manager tidak dipilih

Sekarang tekan tombol putar, dan Anda akan melihat BowlingBall dan Bowling_Pin jatuh hingga tak terhingga (dan sangat mungkin melampaui tak terhingga).

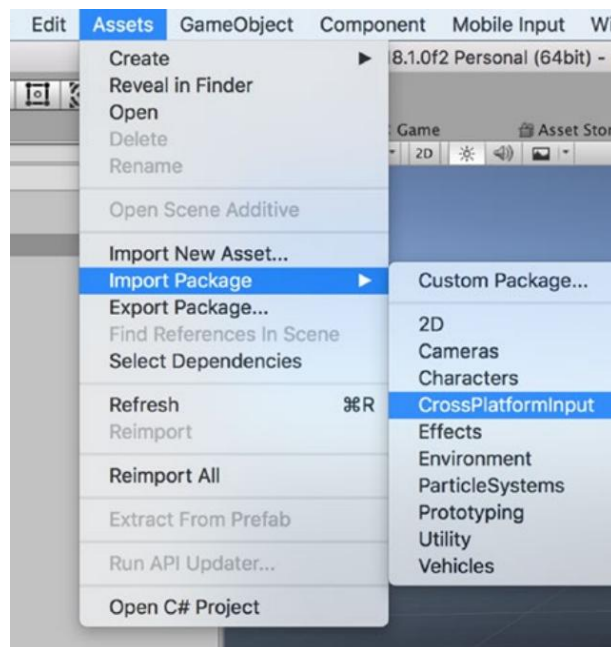
Cara kedua untuk menguji permainan kita adalah dengan membiarkan Unity AR Camera Manager tetap dipilih dan menghubungkan perangkat kita ke Mac, lalu menjalankan Unity AR Kit Remote App di perangkat kita. Kemudian di tab Konsol, pilih perangkat kita dan pilih tombol putar di Unity. Setelah memilih tombol Putar, Anda akan diminta dengan perintah Mulai Sesi Jarak Jauh ARKit, dan pratinjau permainan akan muncul di tab Permainan Unity. Namun, sekali lagi, GameObject BowlingBall dan Bowling_Pin jatuh dan terus jatuh.

6.3 MENERAPKAN KONTROL SENTUH

Sekarang kita akan mencoba membuat Bowling_Ball merespons saat pengguna menyentuh layar pada perangkat mereka. Selama bertahun-tahun, Unity telah membuat sejumlah perubahan pada cara mengelola masukan pengguna menggunakan perangkat dengan kontrol sentuh. Dalam contoh saya, saya akan menggunakan Unity CrossPlatformInput Manager, yang merupakan cara untuk mengelola berbagai bentuk masukan (termasuk sentuhan), saat saya menulis buku ini.

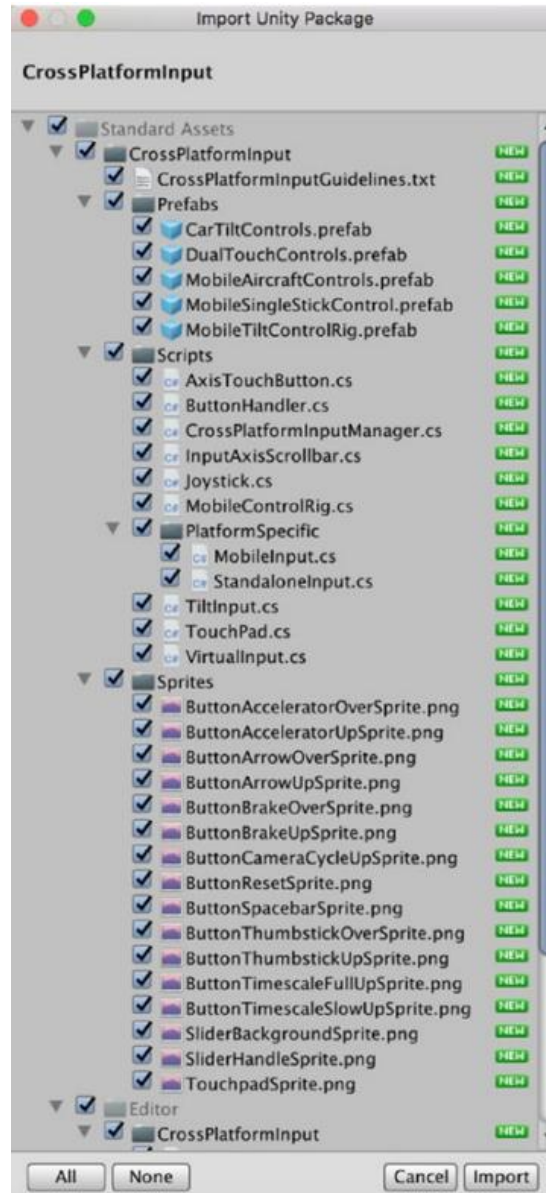
Mengimpor Paket Unity CrossPlatformInput

Mari mengimpor paket Unity CrossPlatformInput. Dari menu, pilih Aset ► Impor Paket ► CrossPlatformInput (Gambar 6.9).



Gambar 6.9. Mengimpor Paket Unity CrossPlatformInput

Berkas akan diunduh dan, setelah selesai, akan membuka utilitas Import Unity Package (Gambar 6.10). Meskipun kita tidak memerlukan semua konten ini, mari pilih Import. Utilitas Import Unity Package sekarang akan menambahkan paket lengkap.



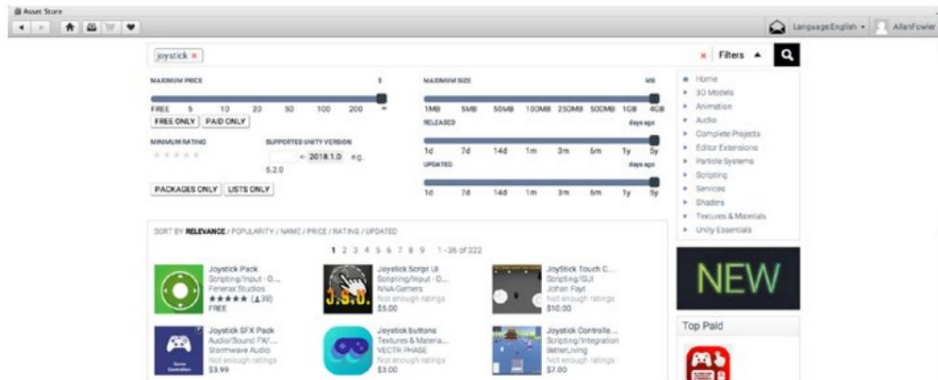
Gambar 6.10. Mengimpor Paket Unity CrossPlatformInput

Menambahkan Kontrol Sentuh

Sekarang kita akan membuat antarmuka pengguna grafis (atau GUI) sehingga pemain dapat mengontrol bola bowling dengan menyentuh layar. Karena kita membuat Fugu Bowl versi AR, kita perlu melakukan hal-hal yang sedikit berbeda. Kita akan membuat pengontrol Joystick dan meletakkannya di layar sehingga pemain dapat menggerakkan bola bowling di sumbu X dan Y. Sekarang, saya tidak banyak melakukan bowling, tetapi saya pun tahu bahwa ini bukanlah cara yang biasa kita lakukan untuk mengontrol bola bowling.

Unduh Paket Aset Joystick

Pertama, kita memerlukan beberapa aset yang bagus, jadi mari kita mulai dan dapatkan sesuatu dari Unity Asset Store. Di Tab Unity Asset store, dari bilah pencarian, cari Joystick (Gambar 6-11). Jika Anda ingin menambahkan filter Free Only, itu mungkin dapat menghemat waktu Anda.



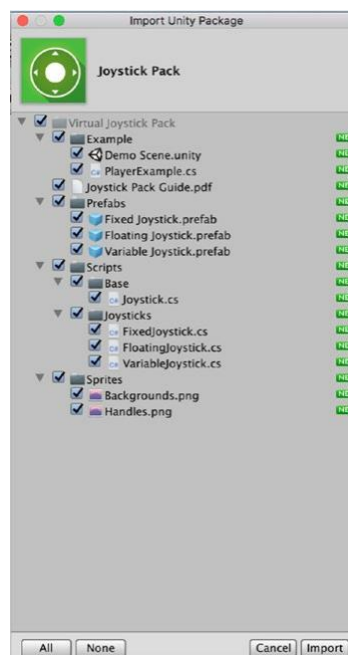
Gambar 6.11. Mencari paket aset Joystick Pack

Pilih paket aset Joy Stick Pack (yang memiliki ikon hijau) dan pilih Impor (Gambar 6.12).



Gambar 6.12. Mengimpor Paket Aset Joy Stick

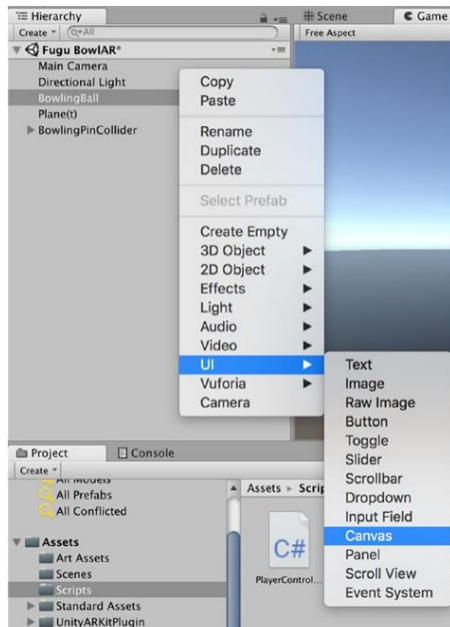
Dari Unity Import Package Utility, pilih Import All (Gambar 6.13).



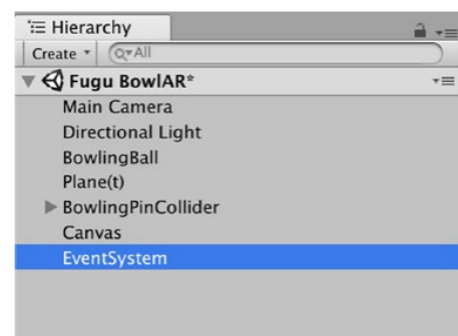
Gambar 6.13. Mengimpor Utilitas Paket Unity

Membuat Kanvas

Di Unity untuk membuat UI, kita perlu membuat kanvas. Di Hierarki, klik kanan dan pilih UI ► Kanvas (Gambar 6.14). Saat kita membuat Kanvas, Unity juga menambahkan GameObject EventSystem (Gambar 6.15). EventSystem saat ini adalah cara Unity menangani peristiwa UI.



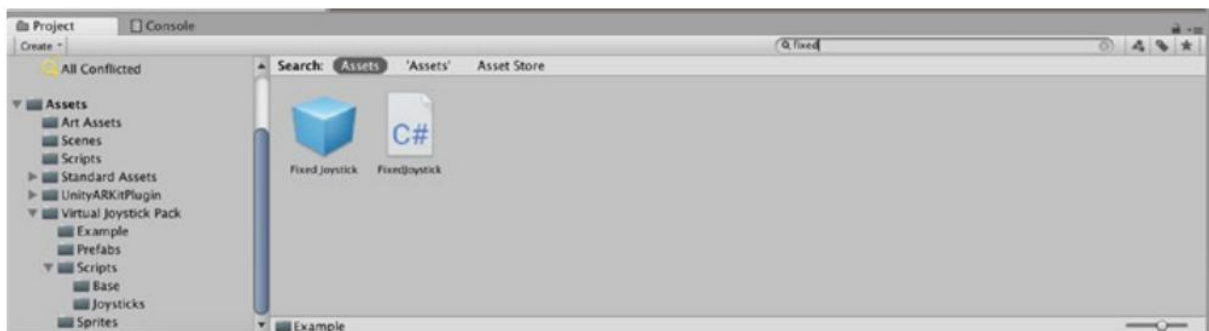
Gambar 6-14. Membuat Kanvas



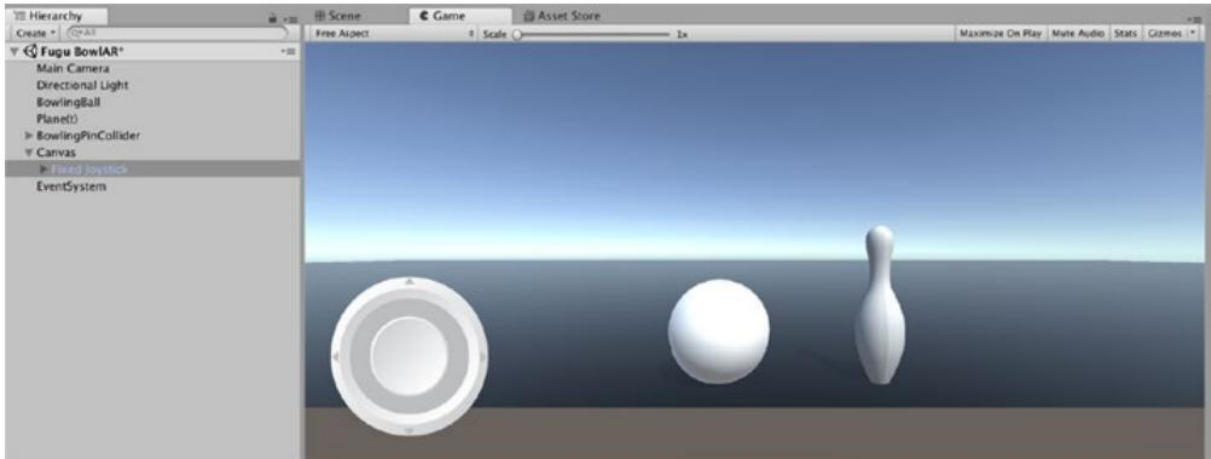
Gambar 6-15. GameObject EventSystem yang ditambahkan ke Hierarki

Menambahkan UI Joystick ke Kanvas

Sekarang kita akan memindahkan Prefab Joystick Tetap dari Folder Joystick Virtual di tab Proyek ke Kanvas. Untuk melakukannya, di tab Proyek, cari Tetap (Gambar 6.16) dan pilih Prefab Joystick Tetap, lalu seret ini ke folder Kanvas di Hierarki (Gambar 6.17). Setelah ditambahkan, Anda benar-benar dapat menguji coba Joystick untuk melihatnya beraksi. Ia tidak akan menggerakkan BowlingBall kita; kita perlu menambahkan skrip untuk melakukannya.



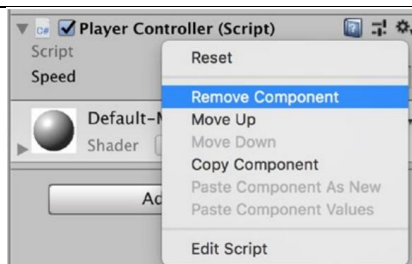
Gambar 6.16. Mencari Prefab Joystick Tetap



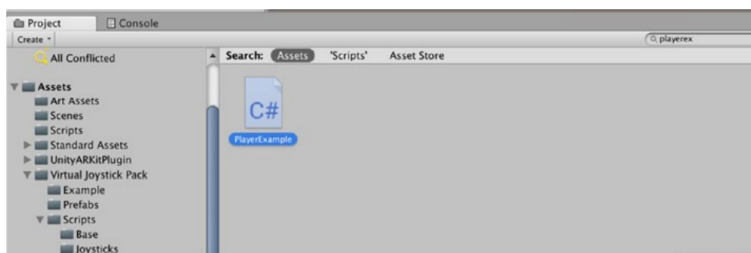
Gambar 6.17. Prefab Joystick Tetap yang ditambahkan ke Canvas

Menambahkan Skrip

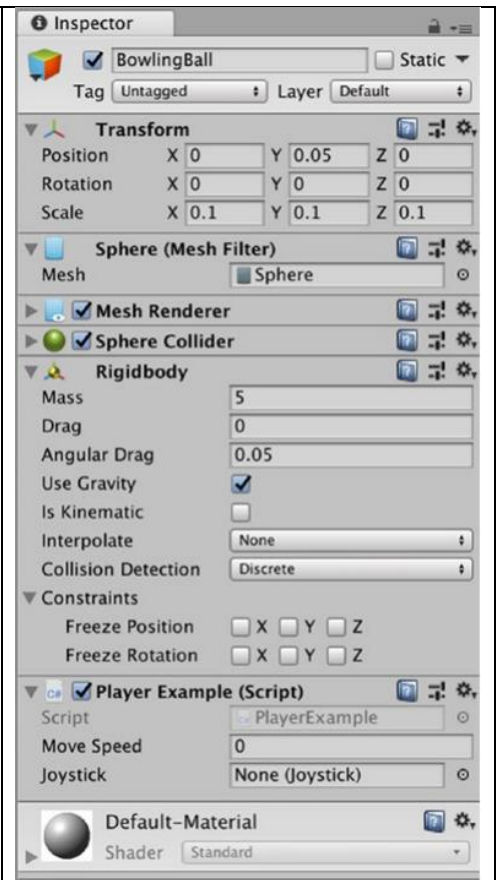
Di Hierarki, pilih GameObject BowlingBall dan di Inspektur, nonaktifkan atau hapus Skrip Pengontrol Pemain. Untuk menghapus Komponen Skrip Pengontrol Pemain, di Inspektur klik kanan roda gigi kecil di sebelah kanan Komponen Skrip Pengontrol Pemain dan pilih Hapus Komponen (Gambar 6-18). Sekarang di tab Proyek, cari dan pilih Skrip PlayerExample (Gambar 6.19) dan seret skrip ini ke Objek Game BowlingBall (Gambar 6.20).



Gambar 6-18. Menghapus Komponen Skrip Pengontrol Pemain

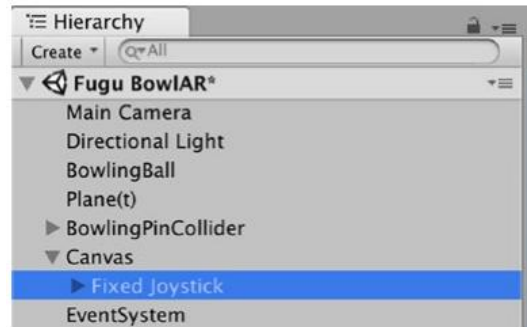


Gambar 6-19. Mencari Skrip PlayerExample

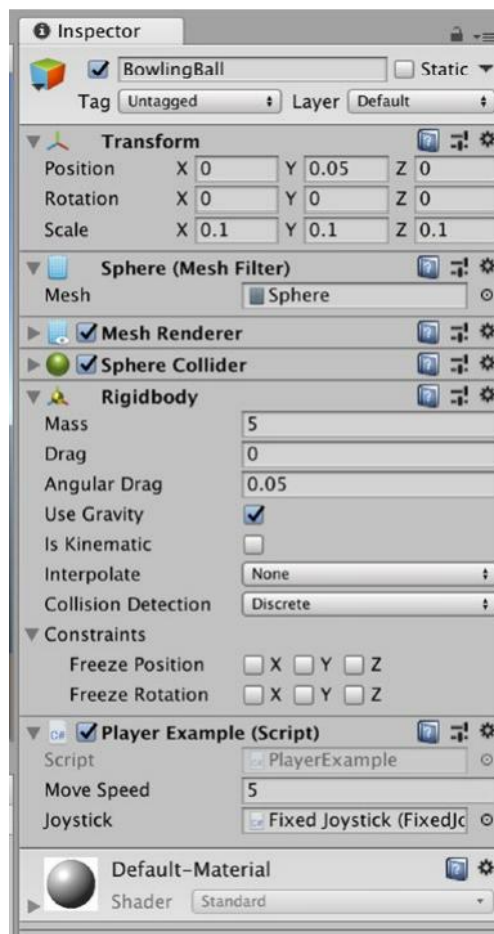


Gambar 6-20. Skrip Player Example yang ditambahkan ke BowlingBall

Sebelum Anda melakukan hal lain, pastikan kecepatannya sudah diatur. Dari Hierarchy, pilih GameObject FixedJoystick (Gambar 6-21) dan seret ini ke kotak properti Joystick dari Komponen Contoh Pemain dari GameObject BowlingBall (Gambar 6.22). Saya telah mengatur Kecepatan ke 5. Ini mungkin agak cepat, tetapi kita dapat mengubahnya nanti.



Gambar 6.21. Memilih GameObject Fixed Joystick



Gambar 6.22. Mengatur properti Kecepatan Gerakan dan Joystick

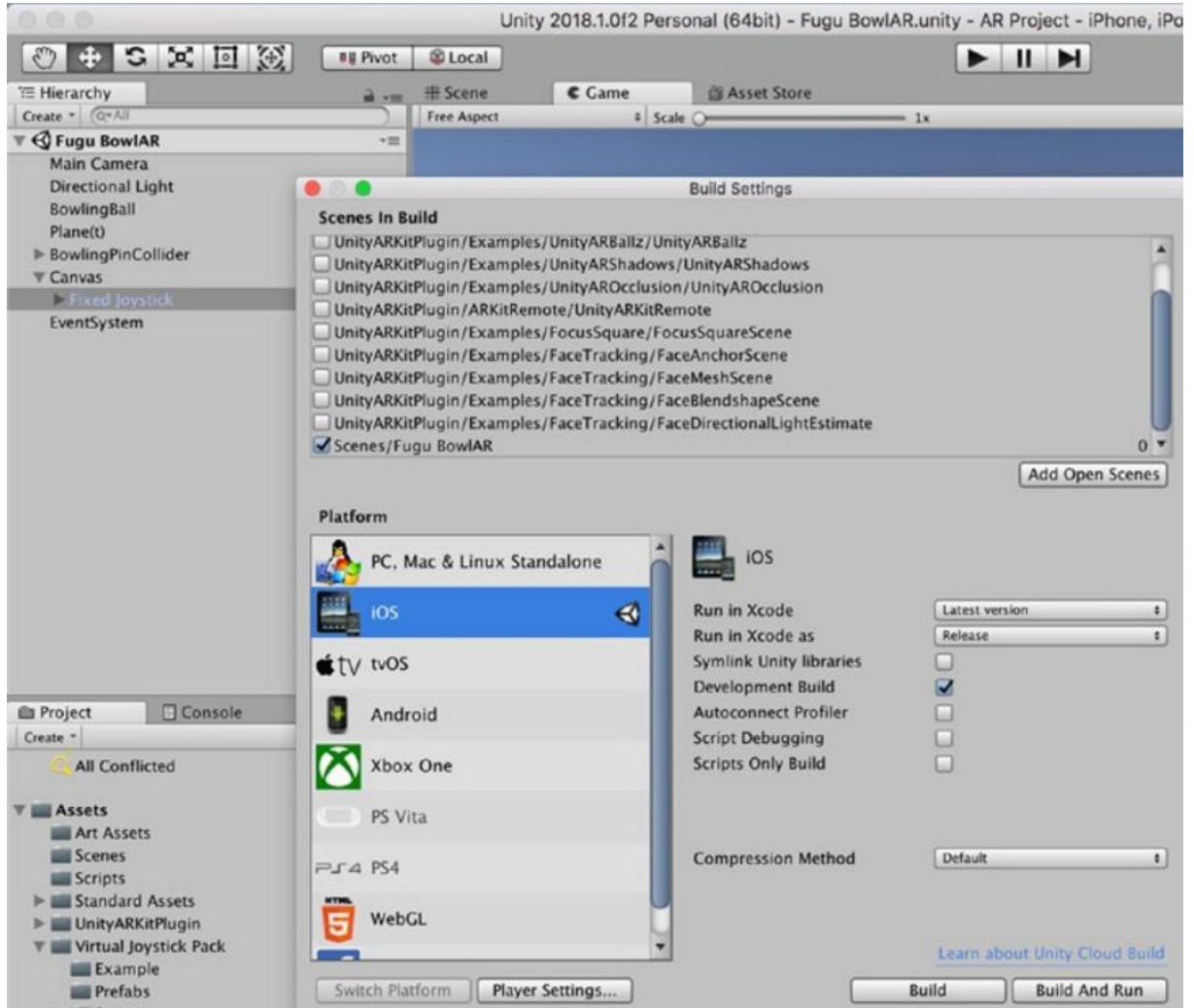
6.4 PENGUJIAN

Sekarang kita dapat menguji Joystick Virtual kita dan melihat apakah berfungsi. Pertama, saya sangat menyarankan Anda mengujinya di tab Game. Ini akan menghemat waktu Anda mengeksport build ke perangkat dan menemukan bahwa build tidak berfungsi (namun, meskipun game akan berfungsi di Unity, ini tidak selalu berarti akan berfungsi di perangkat kita).

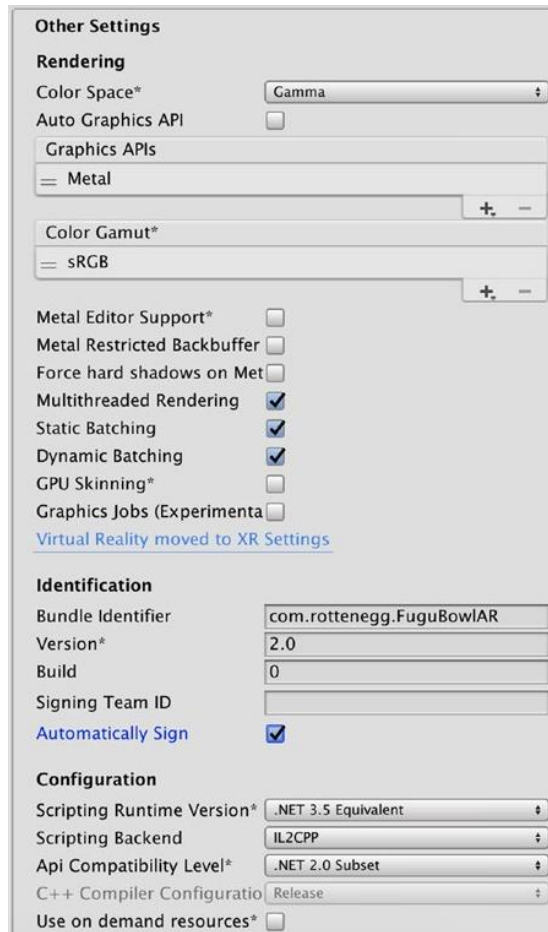
Jika game berfungsi di Unity, sekarang uji di perangkat.

Pada Pengujian Perangkat

Dari menu Unity, pilih File ► Build Settings (Gambar 6.23). Di Build Settings, pilih player settings dan, di Inspector, ubah nomor versi (Gambar 6-24).



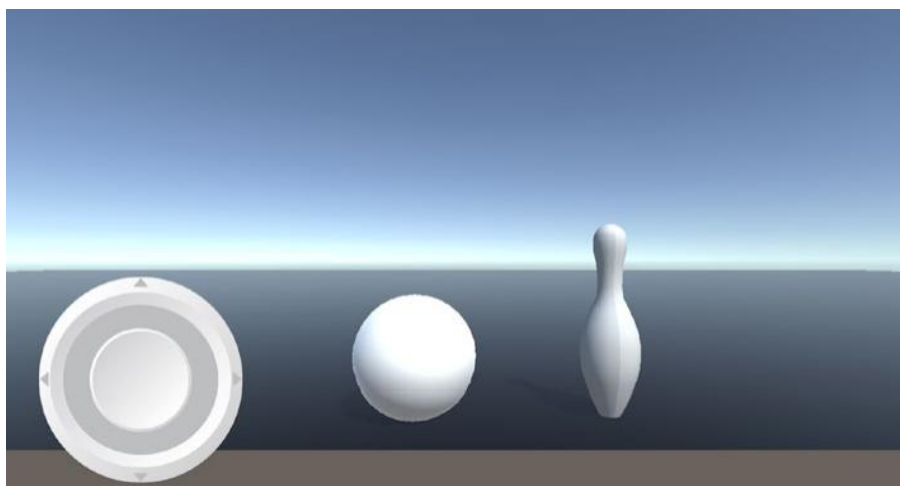
Gambar 6.23. Build Settings



Gambar 6.24. Pengaturan Pemain

Berhasil

Setelah kita memilih Build and Run dari menu Build, Unity akan mengompilasi kode dan membuka Xcode. Setelah kode dikompilasi dalam Xcode, game akan diinstal pada perangkat iOS Anda. Pada Gambar 6-25, saya telah berhasil membuat Aplikasi dan Joystick Virtual berfungsi. Seperti yang disebutkan, kecepatan BowlingBall terlalu cepat, tetapi untuk saat ini, mari kita biarkan apa adanya.



Gambar 6-25. FuguBowAR di iPhone saya

BAB 7

MENAMBAHKAN DETEKSI BIDANG DAN AWAN TITIK

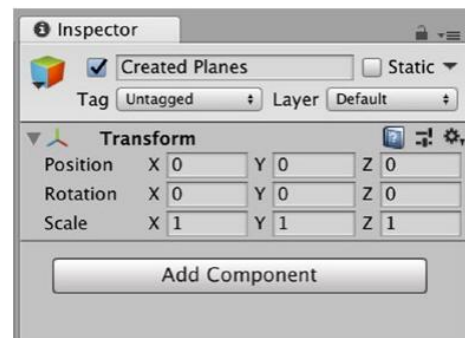
Sekarang kita akan menambahkan Bidang yang Dihasilkan ke permainan kita, yang akan membantu kamera ARKit melacak pergerakan perangkat.

7.1 MEMBUAT GAMEOBJECT BIDANG YANG DIHASILKAN

Sama seperti yang kita lakukan di Bab 3, buat GameObject kosong. Dari menu file, pilih GameObject ► Kosong (Gambar 7.1). Dengan GameObject yang dipilih, di Inspector, beri nama Created Planes (Gambar 7.2).



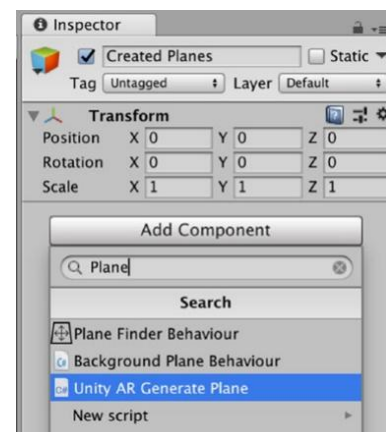
Gambar 7.1. Membuat GameObject kosong



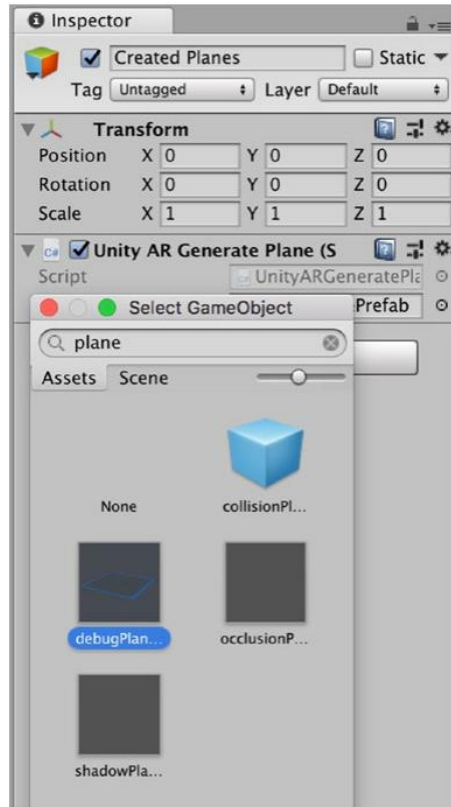
Gambar 7.2. Memberi Nama GameObject Created Planes

Sekarang kita perlu menambahkan komponen. Pilih tombol Add Component di Created Planes Inspector; dan di bilah pencarian, cari skrip Unity AR Generated Plane (Gambar 7.3).

Dengan skrip Unity AR Generated Plane yang disertakan dalam Created Planes GameObject, dalam pengaturan Plane Prefab dari skrip Unity AR Generated Plane, pilih roda gigi kecil di sebelah kanan kotak opsi dan cari serta pilih DebugPlanePrefab (Gambar 7.4).



Gambar 7.3. Mencari skrip Unity AR Generated Plane



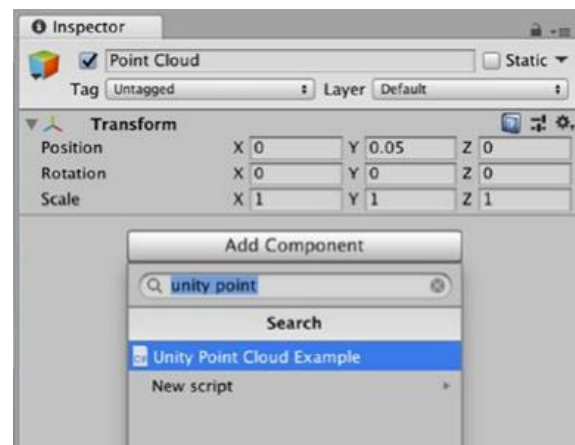
Gambar 7.4. Mencari debugPlanePrefab

Membuat GameObject Point Cloud

Buat GameObject kosong (Gambar 7.5). Dengan GameObject Kosong yang dipilih di Inspektur, ubah namanya menjadi Point Cloud. Sekarang tambahkan komponen. Di bilah pencarian, cari Unity Point Cloud Particle Example dan tambahkan ini ke GameObject Point Cloud (Gambar 7.6).



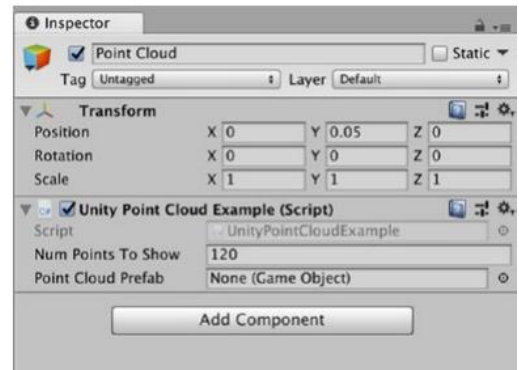
Gambar 7.5. Membuat GameObject kosong



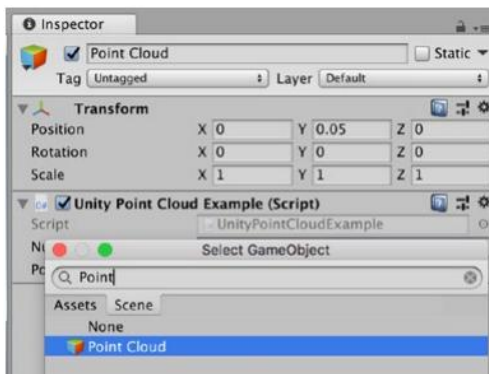
Gambar 7.6. Mencari Contoh Partikel Awan Titik

Dengan skrip Contoh Partikel Awan Titik Unity yang ditambahkan, sekarang atur jumlah Titik Maksimum yang akan Ditampilkan menjadi 120 (Gambar 7.7). Anda dapat mengatur sebanyak mungkin Awan Titik sesuai keinginan. Sekarang tambahkan Point Cloud Particle Prefab ke kotak properti Point Cloud Prefab. Pilih roda gigi kecil di sebelah kanan kotak properti Point Cloud Prefab, lalu cari Point Cloud Prefab (Gambar 7.8).

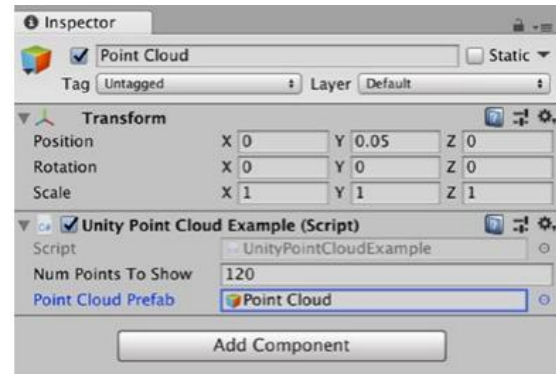
Pilih dan seret PointCloudPrefab ke kotak Point Cloud Particle Prefab di skrip Contoh Point Cloud Particle (Gambar 7.9).



Gambar 7.7. Mengatur Titik Maksimum yang akan Ditampilkan



Gambar 7.8. Mencari PointCloud Prefab



Gambar 7.9. Set PointCloudPrefab sebagai Point Cloud Prefab

7.2 MENYIAPKAN KAMERA UTAMA

Anda mungkin ingat bahwa di Bab 3, kita perlu menyiapkan permainan untuk AR. Hal pertama yang perlu kita lakukan adalah mengubah pengaturan Kamera Utama. Jika Anda ingat cara melakukannya, Anda dapat melewati bagian ini.

Namun, saya akan memberikan panduan untuk menyiapkannya bagi para pembaca yang membaca Bab 3 beberapa waktu lalu dan mungkin memerlukan pengingat tentang cara kita menyiapkannya. Di Hierarki, pilih Kamera Utama dan di Inspektur, atur Clear Flags ke Depth only (Gambar 7.10).



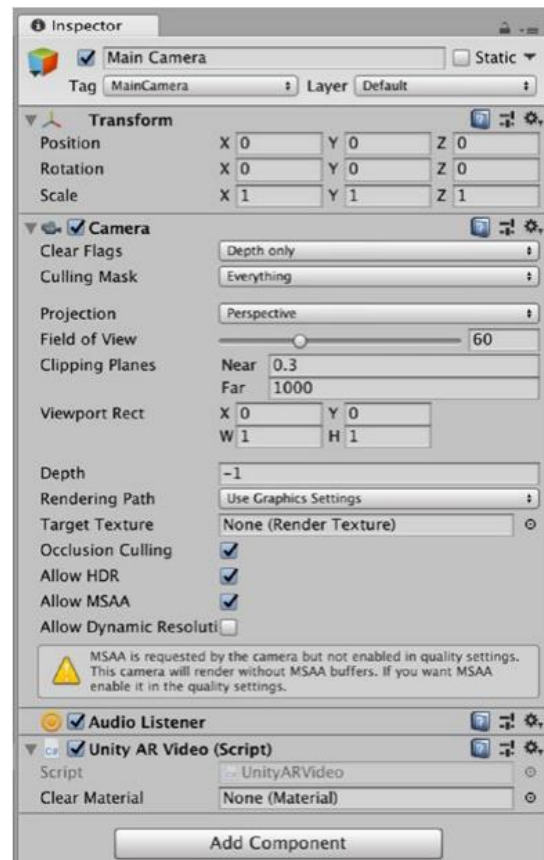
Gambar 7-10. Atur Clear Flags ke Depth only

Dengan Kamera Utama dipilih, di Inspektur pilih Tambahkan Komponen.

Sekarang cari dan tambahkan Skrip Video AR Unity (Gambar 7-11). Sekarang pilih Unity AR Video Script (klik kiri sekali) dan komponen ini akan ditambahkan ke Kamera Utama (Gambar 7-12).



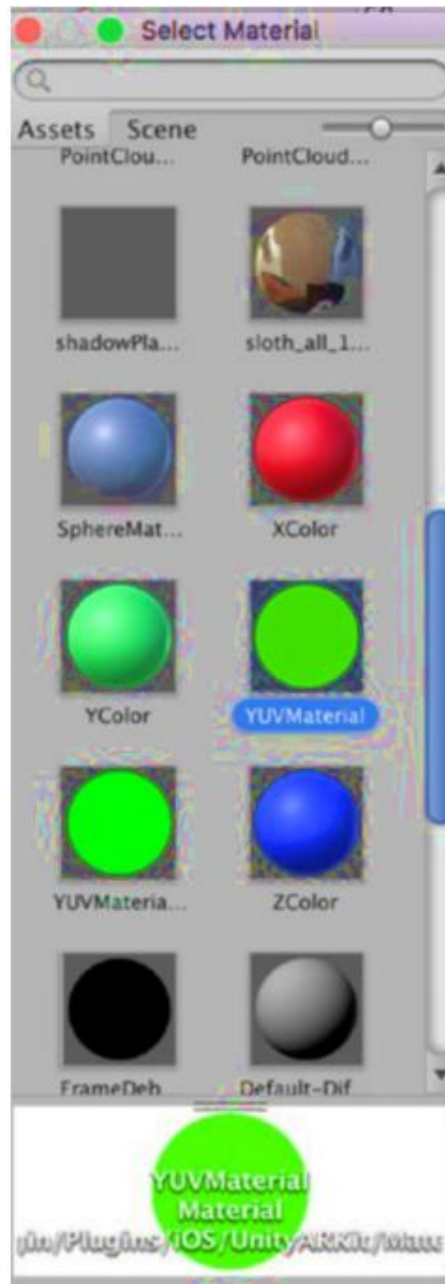
Gambar 7-11. Mencari komponen Skrip Video AR Unity



Gambar 7-12. Komponen Unity AR Video Script yang ditambahkan ke kamera utama

Mengatur Material Clear pada Unity AR Video Script

Di sebelah kanan kotak properti komponen Unity AR Video Script, terdapat roda gigi kecil (Gambar 7.12); pilih roda gigi ini, dan cari material YUV (Gambar 7.13).



Gambar 7-13. Memilih material YUV untuk Clear Material

Menambahkan Unity AR Camera Manager

Dengan Kamera Utama yang masih dipilih, di Inspector, pilih Add Component (Gambar 7-14); sekarang cari AR Camera Manager (Gambar 7-15), dan sekarang tambahkan ke Kamera Utama.

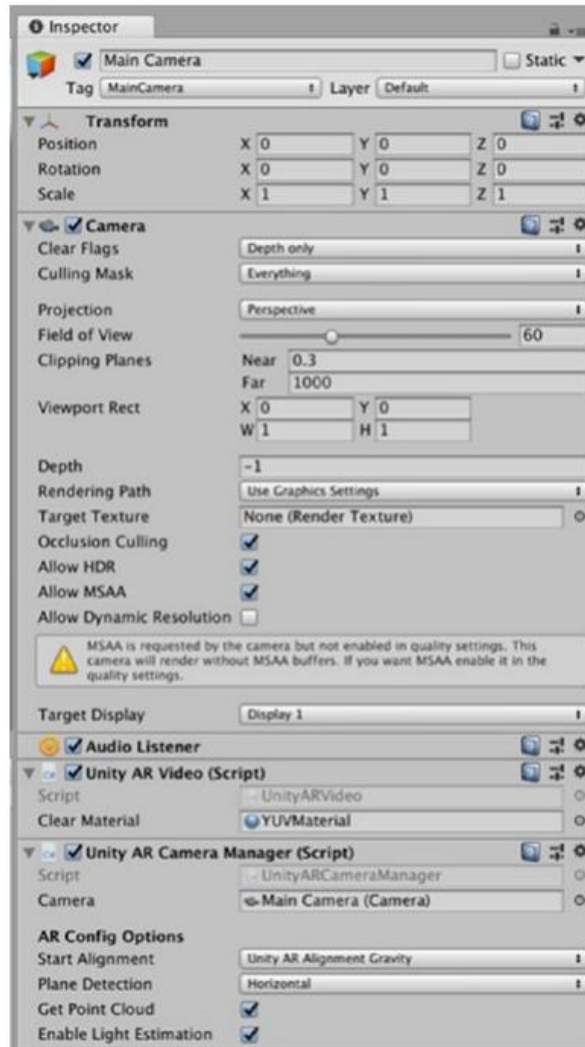


Gambar 7.14. Tambahkan Komponen



Gambar 7.15. Mencari Unity AR Camera Tracker

Sekarang tambahkan Kamera Utama sebagai Kamera Terlacak. Untuk menambahkan Kamera Utama ke Properti Kamera Unity AR Camera Manager, pilih Kamera Utama dari Hierarki dan seret ini ke properti Kamera Terlacak Unity AR Camera Manager (Gambar 7.16).



Gambar 7.16. Menetapkan Kamera Utama sebagai Kamera Terlacak

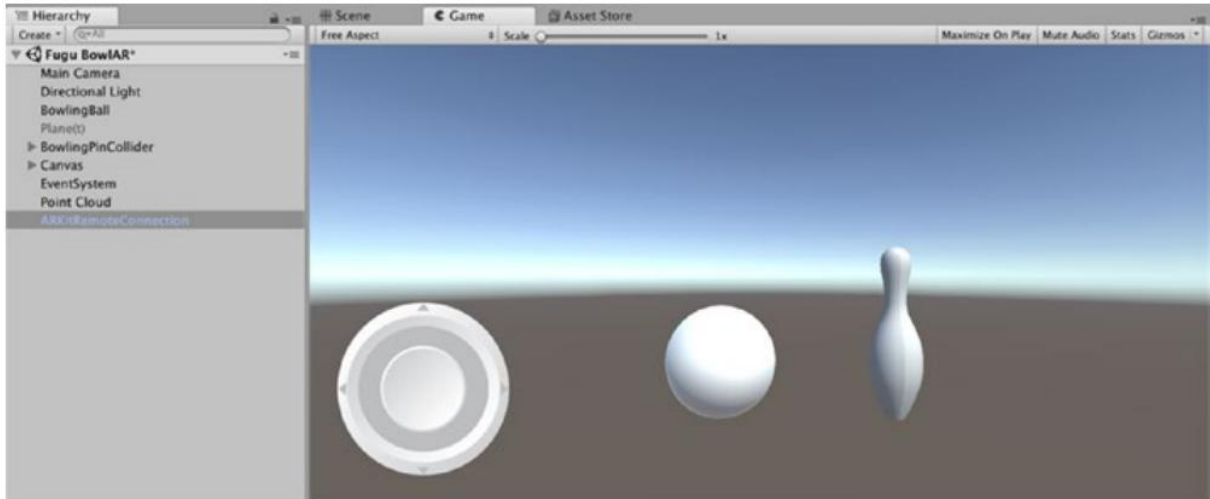
7.3 MENAMBAHKAN UNITY REMOTE CONNECTION

Untuk menguji permainan kita di tab Permainan, kita juga perlu menambahkan Unity Remote Connection ke Hierarki kita. Di Tab Proyek, cari Prefab yang disebut ARKitRemoteConnection (Gambar 7.17).



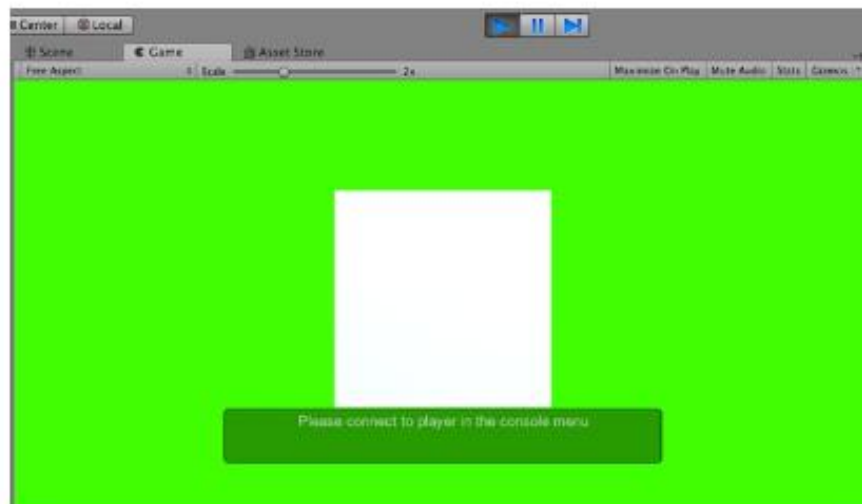
Gambar 7.17. Mencari Prefab UnityARKitRemoteConnection

Pilih Prefab ini dan tambahkan ke Scene (seret ke Hierarchy) (Gambar 7.18).



Gambar 7.18. UnityARKitRemoteConnection di Scene

Sekarang kita perlu membuka ARKitRemote di perangkat iOS kita. Pertama, hubungkan perangkat iOS Anda ke komputer pengembangan Anda. Di Unity, pilih tab Game dan tekan tombol Play. Unity akan meminta Anda untuk terhubung ke pemain di menu konsol (Gambar 7-19).



Gambar 7.19. Pesan Unity connect to player

Di tabel Console, pilih Editor dan pilih perangkat iOS Anda (Gambar 7.20).



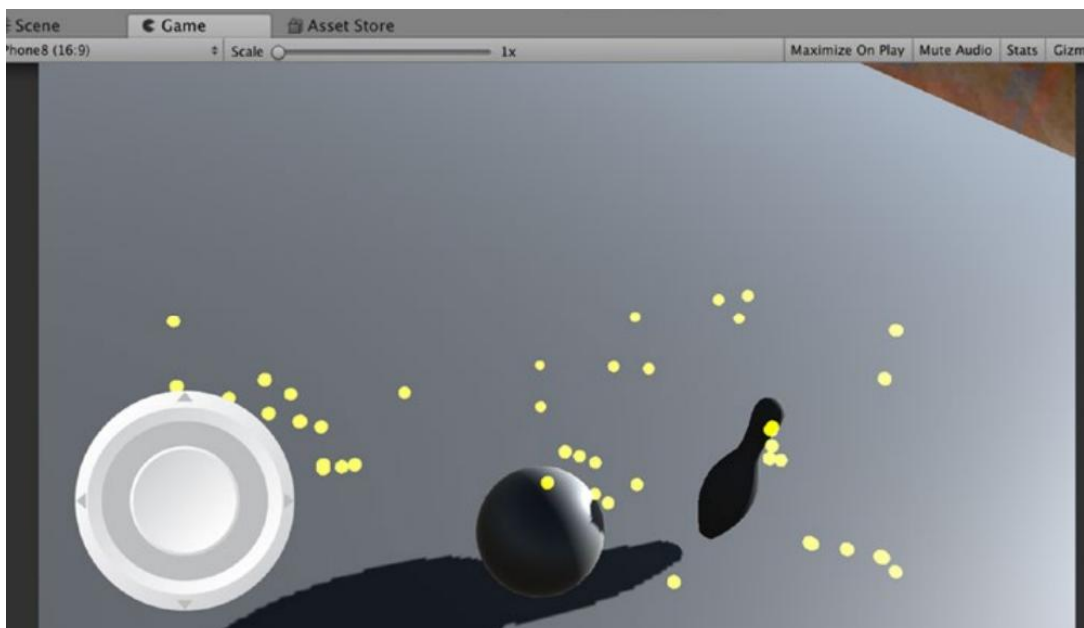
Gambar 7.20. Memilih perangkat iOS di menu Console

Kemudian Unity akan meminta Anda untuk Start Remote ARKit Session (Gambar 7.21). Klik ikon di layar Game di Unity dan aplikasi Anda sekarang dapat berjalan di perangkat iOS Anda.



Gambar 7.21. Perintah Mulai Sesi ARKit Jarak Jauh

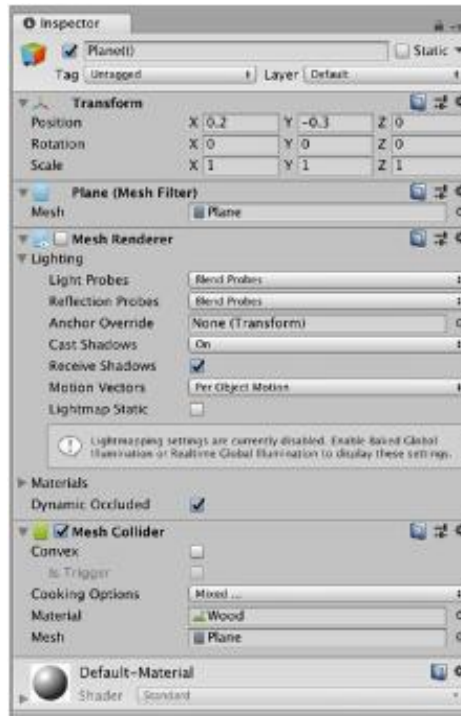
Pada Gambar 7.22 Anda dapat melihat bahwa saya telah berhasil melihat Aplikasi Hello WorldAR di iPhone saya dan melihat Scene di Unity. Jika Anda menggerakkan perangkat, Anda akan melihat Kamera bergerak di tab Unity Scene dan gambar berubah di tab Game. Saya ingin menunjukkan bahwa prosesnya akan sedikit lambat (lag), tetapi ARKitRemote saat ini merupakan cara terbaik untuk menguji pengembangan Unity AR untuk iOS.



Gambar 7.22. Tampilan Game

Houston, Kita Punya Masalah...

Jika Anda mendapatkan hasil yang sama seperti yang saya dapatkan pada Gambar 7.23, jangan panik! Adegan tersebut dirender secara real time. BowlingBall dan Bowling_Pin sedang dirender dan begitu pula Virtual Joystick. Kita dapat melihat Point Clouds. Namun... Pane(t) masih terlihat.



Gambar 7.23. Mesh Renderer tidak dipilih

Untuk memperbaiki tantangan ini, kita akan menonaktifkan Mesh Renderer Component dari Plane(t) kita. Untuk melakukannya, di Hierarchy, pilih Plane(t) GameObject dan di Inspector, batalkan pilihan Mesh Renderer (Gambar 7-24). Untuk melakukannya, klik kiri mouse pada tanda centang di kotak centang di sebelah kiri Mesh Renderer Component. Sekarang uji ulang build Anda dan lihat hasilnya. Hasil Anda seharusnya mirip dengan hasil saya (Gambar 7-24).

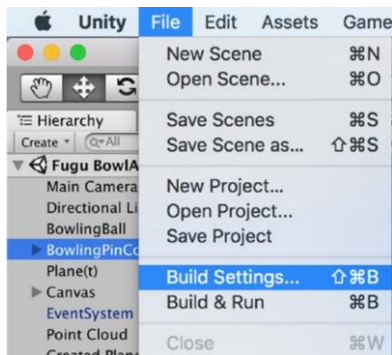


Gambar 7.24. Tampilan Game tanpa Plane(t)

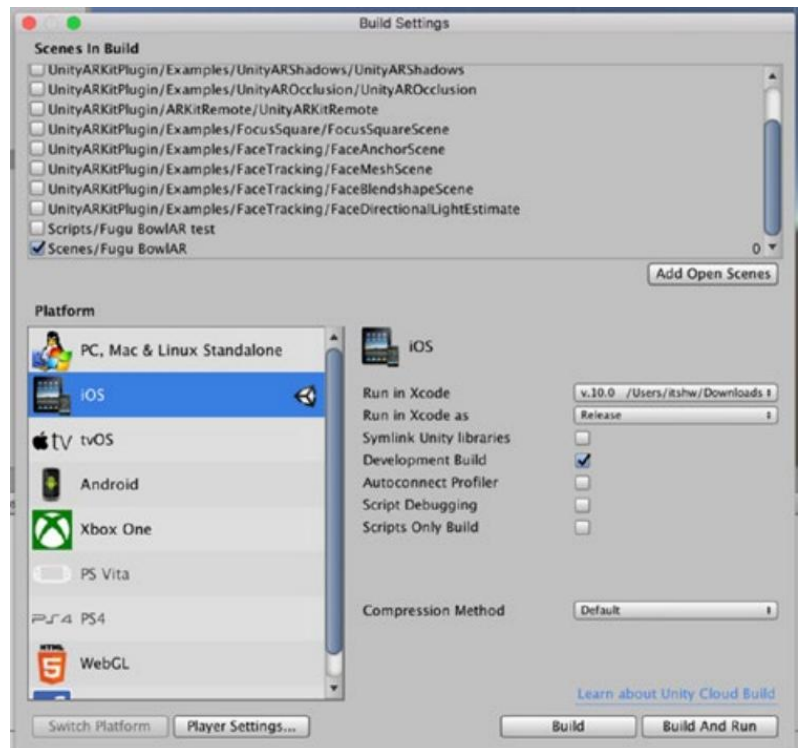
7.4 PENGUJIAN PADA PERANGKAT IOS

Karena kami telah berhasil menguji build terkini di Tampilan Game, sekarang adalah saat yang tepat untuk melihat apakah ini berfungsi di perangkat kami.

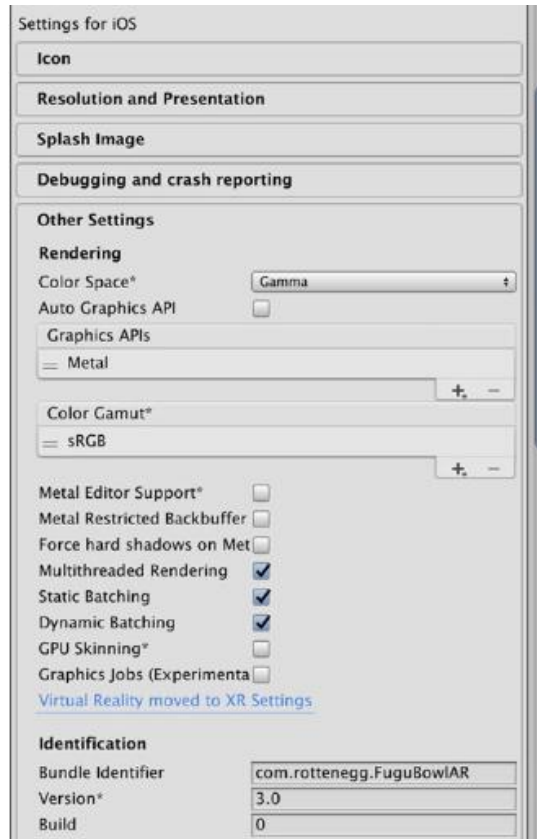
Dari Menu Utama, pilih dari Menu Utama, File ► Pengaturan Build (Gambar 7-25). Pastikan Anda telah memilih iOS sebagai Platform, dan kotak centang Build Pengembangan dipilih. Pastikan juga untuk memeriksa apakah Anda membangun adegan yang benar (Gambar 7-26). Di Menu Pengaturan Build, pilih Pengaturan Pemain dan di Inspektur pastikan bahwa opsi Pengaturan Lainnya ditetapkan dengan benar. Saya telah menetapkan versi saya ke versi 3.0. Namun, Anda mungkin ingin menggunakan nomor versi yang berbeda (Gambar 7-27).



Gambar 7.25. Memilih Pengaturan Build



Gambar 7.26. Menu Pengaturan Build



Gambar 7.27. Opsi Menu Pengaturan Lainnya

Dari menu Pengaturan Build, pilih Build dan Run. Unity akan mengompilasi kode lalu membuka Xcode. Pastikan perangkat iOS Anda terhubung dan Anda telah menyetel Provisioning Profile di Xcode dengan benar. Jika semuanya disetel dengan benar, Xcode akan menyebarkan game Anda ke perangkat iOS, dan Anda dapat menguji build tersebut.

Menyeimbangkan Kekuatan

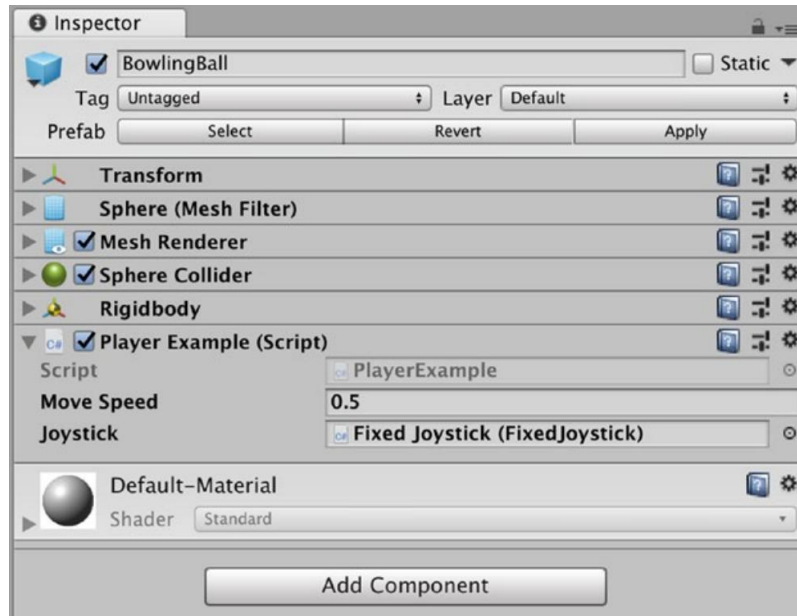
Saat kami menguji permainan kami dalam tampilan Permainan, kami menemukan bahwa Objek Permainan BowlingBall mungkin berjalan agak terlalu cepat. Jika Anda dapat menguji permainan di perangkat iOS Anda, Anda mungkin merasa bahwa kami memiliki situasi yang sama. Proses membuat permainan yang adil atau seimbang tidaklah mudah; hal itu memerlukan banyak perencanaan dan pengujian. Dalam permainan kami, kami menetapkan Objek Permainan BowlingBall pada kecepatan 5 m/jam. Seperti yang mungkin Anda bayangkan, pemain bowling rata-rata tidak mungkin dapat melempar bola bowling pada kecepatan 5 m/jam. Namun, karena ini adalah permainan, kami memiliki sedikit kebebasan dan kebebasan kreatif. Namun, kami tidak ingin membuat permainan terlalu sulit atau terlalu mudah. Menemukan keseimbangan ini merupakan tantangan. Seperti semua acara memasak yang hebat, saya akan menguji pengaturan dan melaporkan hasilnya.

Namun, jika Anda ingin mencoba menguji kecepatan yang berbeda untuk bola bowling, ini akan menjadi pengalaman yang hebat. Namun, saya akan merekomendasikan untuk menguji permainan dalam tampilan Permainan dan tidak menerapkan build ke perangkat iOS

Anda.

Memperlambat Bola Bowling

Ada dua cara untuk memperlambat bola bowling.



Gambar 7-28. Pengaturan Kecepatan Gerak yang disesuaikan

Kita bisa menambah massanya (membuatnya lebih berat) atau memperlambat kecepatan gerakannya. Saya lebih suka opsi kedua. Di Hierarchy, pilih BowlingBall GameObject dan di Inspector atur Move Speed di komponen Player Example menjadi 0,5 (Gambar 7-28). Saya cukup puas dengan pengaturan ini, tetapi beberapa pengujian pengguna akan membantu kami mengonfirmasi apakah kecepatan ini terlalu lambat.

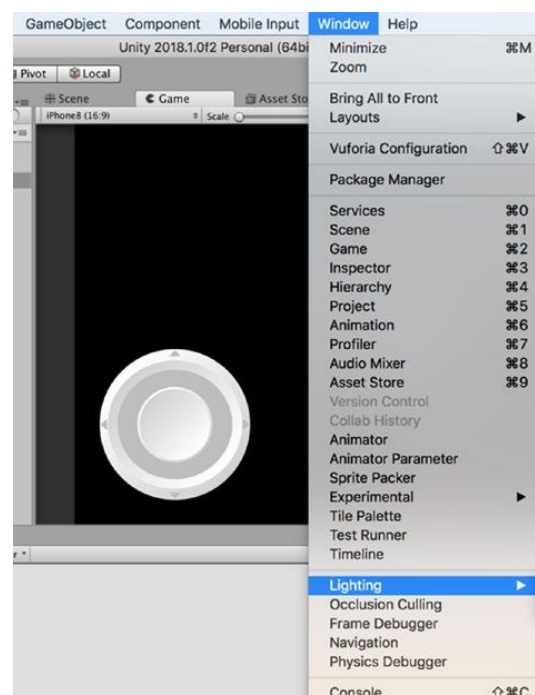
7.5 PENCAHAYAAN

Sama seperti yang kami lakukan di Bab 4, kami akan menyesuaikan pencahayaan sehingga lampu dalam permainan mendekati pencahayaan di dunia nyata.

Mematikan Lampu

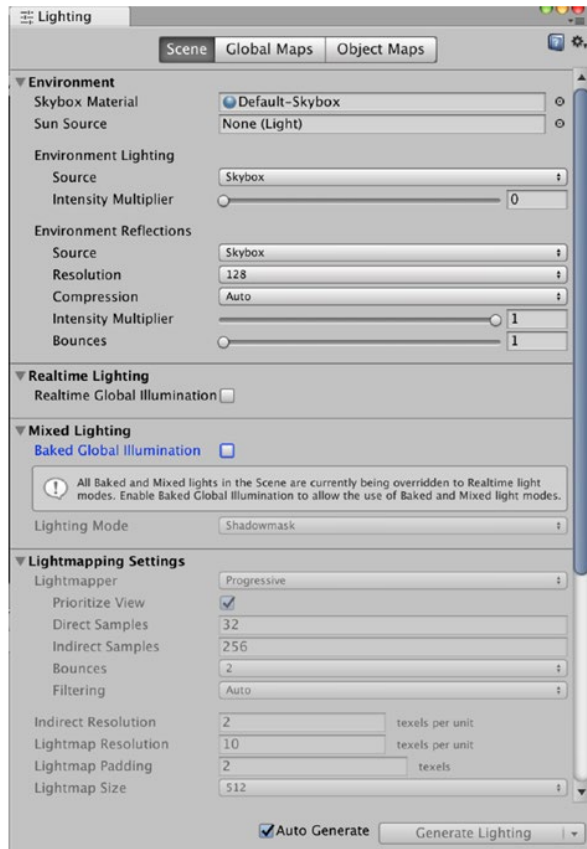
Kita akan menggunakan sumber cahaya yang berbeda untuk Aplikasi kita. Namun, untuk hasil terbaik, pastikan kita telah mematikan semua lampu di tempat kejadian.

Dari Menu Utama, pilih Jendela ► Pencahayaan ► Pengaturan (Gambar 7-29).

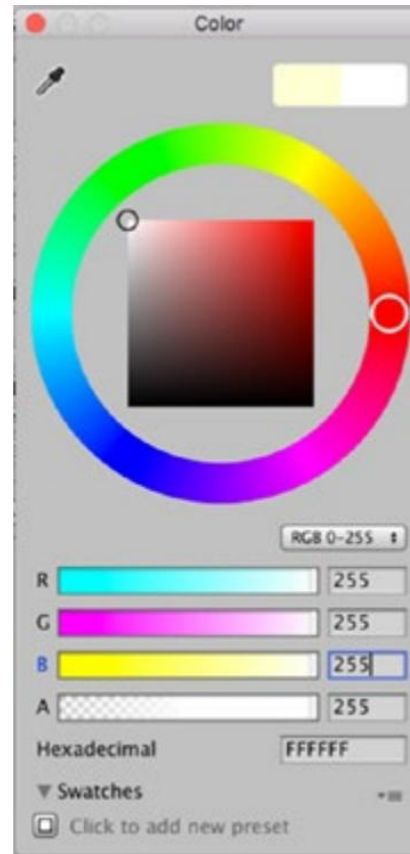


Gambar 7-29. Memilih Pengaturan Pencahayaan

Pada Pengaturan Pencahayaan, pada pengaturan Lingkungan, ubah Pengganda Intensitas menjadi nol. Pada pengaturan Pencahayaan Campuran, hapus centang pada pengaturan Pencahayaan Terpanggang (Gambar 7-30). Sekarang pilih Directional Light dan di Inspector, ubah warnanya menjadi putih. Lalu ketik nilai RGB atau HSV untuk warna yang Anda inginkan di kotak properti (Gambar 7-31).



Gambar 7-30. Mengubah Pengaturan Pencahayaan

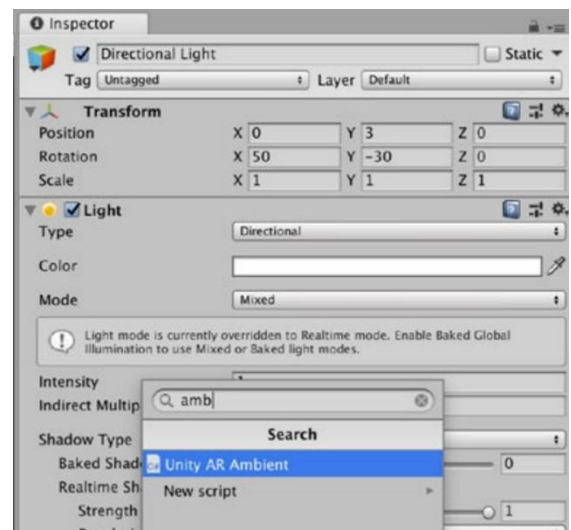


Gambar 7-31. Mengetik di properti warna

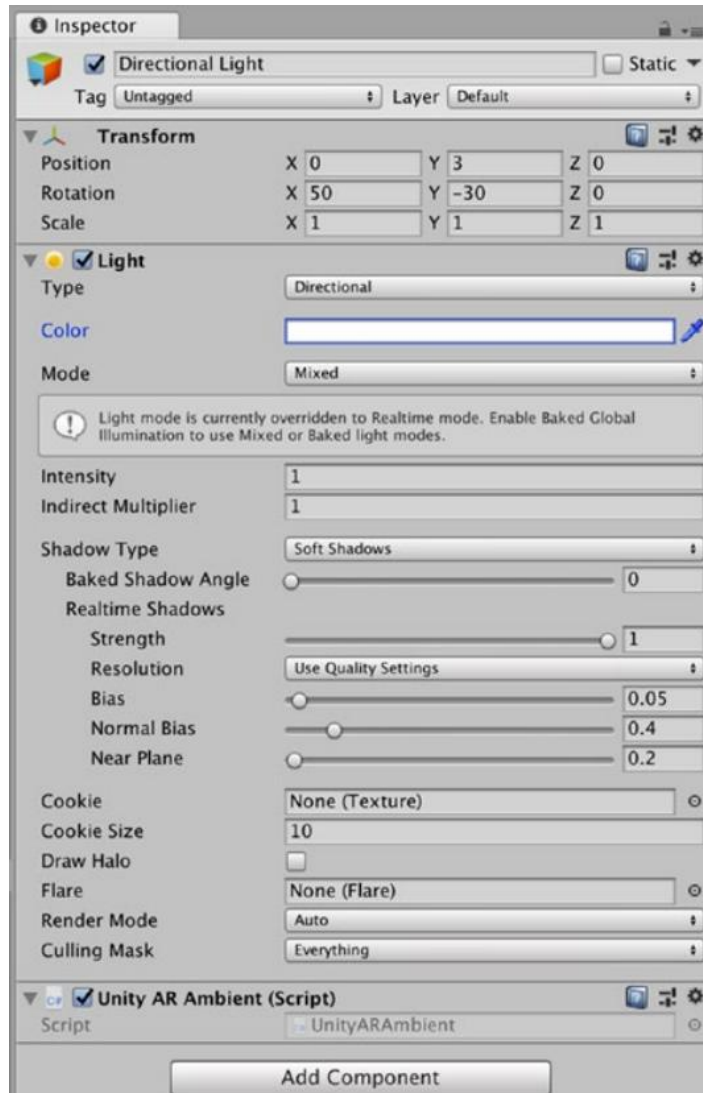
Menetapkan Sumber Cahaya Sekitar

Sekarang kita dapat menetapkan sumber cahaya sekitar. Dengan memilih Cahaya Terarah, pilih tombol Tambahkan Komponen dan cari ambient (Gambar 7-32).

Anda akan melihat bahwa Unity ARKit juga menyertakan skrip yang disebut Unity AR Ambient. Pilih skrip ini untuk menambahkannya ke Directional Light (Gambar 7-33).



Gambar 7-32. Mencari Unity AR Ambient



Gambar 7-33. Skrip Unity AR Ambient yang ditambahkan ke Directional Light

Ringkasan

Dalam bab ini, kami menambahkan Plane Detection dan Point Clouds ke Game kami. Kami juga menyempurnakan pencahayaan agar pencahayaan dalam game tampak lebih mirip dengan pencahayaan di dunia nyata. Dalam bab berikutnya, kami akan membahas cara menambahkan Suara ke game kami.

Pengembang yang lebih berpengalaman yang membaca bab ini mungkin akan menyadari bahwa kami masih memiliki GameObject Plane(t) di dalam Game. Jika kami menghapus GameObject Plane(t) sepenuhnya, BowlingBall dan Bowling_Pin akan terus jatuh. Ini karena GameObject ini dibuat sebelum Created Plane dibuat. Dalam Bab 8, kami akan membahas cara membuat GameObject ini di Unity setelah Created Plane dibuat. Untuk saat ini, kami memiliki lingkungan pengujian yang cukup solid.

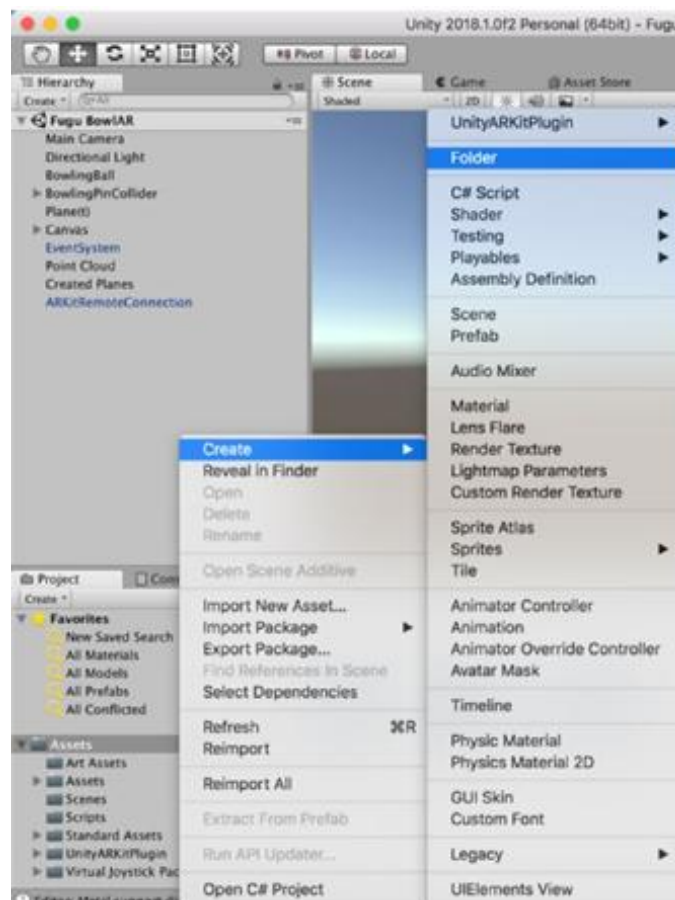
BAB 8

LANGKAH AKHIR

Pada bab sebelumnya, kita menemukan bahwa untuk menghentikan bola bowling dan pin agar tidak jatuh melalui bidang yang kita buat, kita perlu menjaga GameObject Plane(t) di dalam adegan. Alasannya adalah Unity membuat (atau membuat instance) bola bowling dan pin GameObject sebelum membuat bidang yang dibuat. Pada bab ini, kita akan membuat beberapa perubahan pada proyek kita agar dapat melakukan ini.

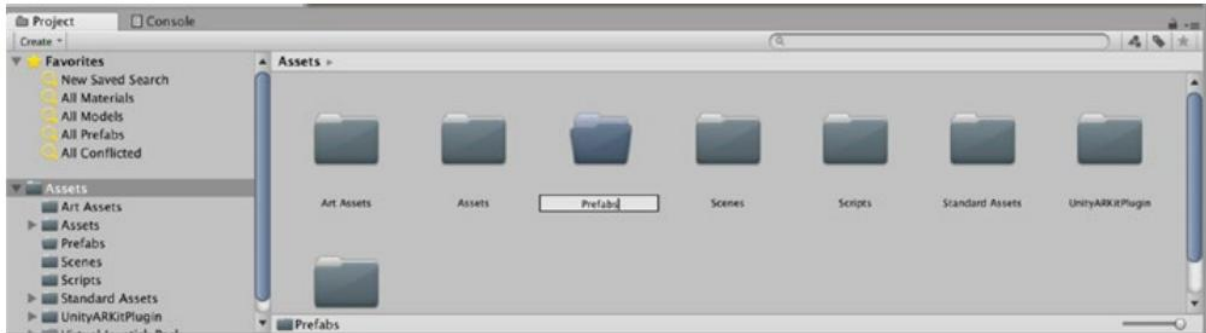
8.1 MEMBUAT PREFAB

Langkah pertama yang perlu kita ambil adalah mengubah GameObject BowlingBall dan Bowling_Pin menjadi prefab. Untuk melakukannya, kita perlu membuat subfolder Prefab di folder Assets. Untuk melakukannya, pilih folder Assets di Tab Project dan klik kanan lalu pilih Create Folder (Gambar 8.1).



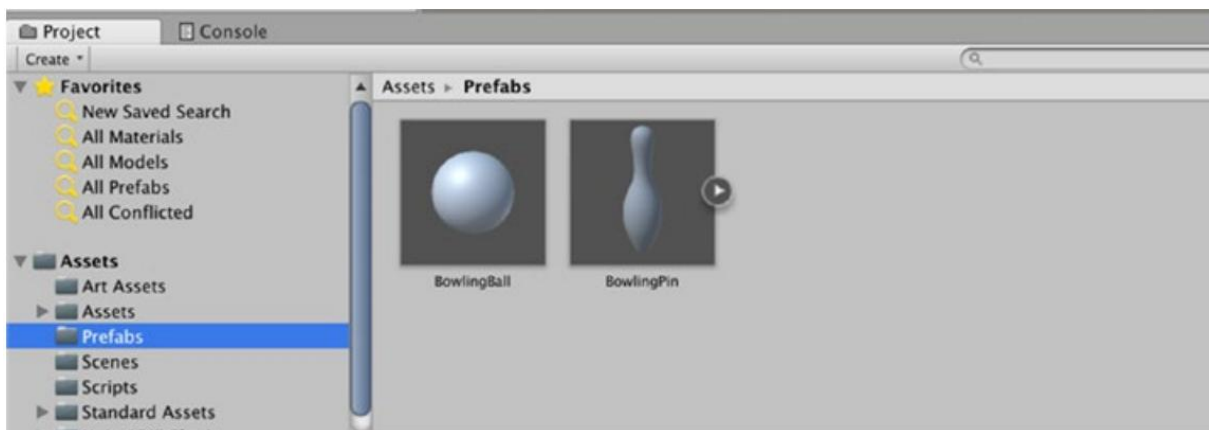
Gambar 8.1. Membuat subfolder di folder Aset

Setelah subfolder baru dipilih, ganti namanya menjadi Prefab (Gambar 8.2).



Gambar 8.2. Membuat folder Prefab

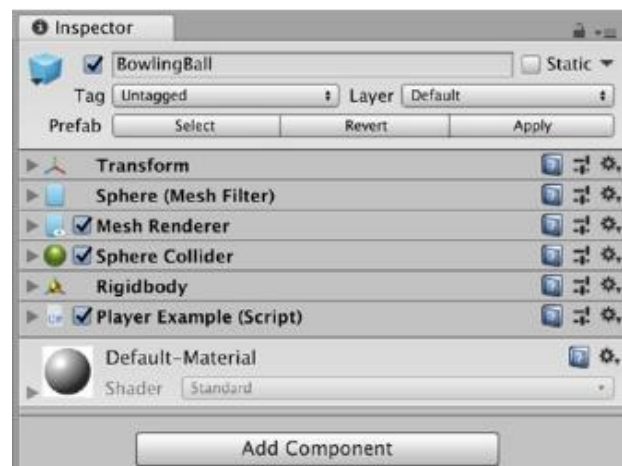
Pertama, saya akan mengganti nama GameObject BowlingPinCollider saya menjadi BowlingPin. Untuk melakukannya di tab Hierarchy, pilih GameObject BowlingPinCollider dan klik kanan objek permainan tersebut dan pilih Rename (Anda juga dapat memilih GameObject di Hierarchy lalu klik kiri nama file tersebut). Sekarang ganti nama file ini menjadi BowlingPin. Ini tidak hanya akan menjaga konvensi penamaan file kita tetap konsisten, tetapi juga akan membantu dalam mengidentifikasi prefab saat dibuat. Sekarang di tab Hierarchy, pilih GameObject BowlingBall dan BowlingPin dan seret keduanya ke folder Prefab (Gambar 8-3).



Gambar 8.3. Prefab BowlingBall dan Bowling Pin

Sekarang Unity telah mengonversi GameObject ini menjadi Prefab dengan baik, dan sekarang kita dapat menyertakannya untuk membuat aset yang baru dibuat ini saat runtime. Untuk memeriksa apakah aset ini sekarang menjadi prefab, pilih BowlingPin di Hierarchy dan di Inspector, Anda akan melihat bahwa properti telah berubah (Gambar 8.4).

Pada Gambar 8.4, lihat bagian atas Inspektur (tepat di bawah properti Tag), dan Anda akan melihat bahwa kita sekarang



Gambar 8.4. Properti prefab BowlingBall

memiliki Properti Prefab (Pilih, Kembalikan, Terapkan). Ini mengonfirmasi bahwa kita telah berhasil mengubah GameObject ini menjadi Prefab. Sekarang, kita dapat membuat instance prefab ini saat runtime. Manfaat tambahannya adalah kita juga dapat membuat instance prefab sebanyak yang kita inginkan. Untuk membuat instance objek game ini saat runtime, kita perlu membuat beberapa kode.

8.2 MENAMBAHKAN LEBIH BANYAK PIN BOWLING

Langkah pertama adalah kita akan menambahkan lebih banyak pin bowling. Untuk melakukannya, kita akan memperbarui Prefab BowlingPin kita. Di Hierarki, pilih GameObject Bowling_Pin dan duplikat sembilan kali. Sekarang kita akan mengganti nama duplikat ini dan mengubah posisi awalnya. Jika Anda melihat pada tampilan Scene atau Game View, Anda akan melihat bahwa semua salinan yang kita buat diposisikan dengan baik pada posisi transformasi yang sama. Meskipun ini mungkin bagus untuk pemain, kita ingin membuatnya sedikit lebih menantang.

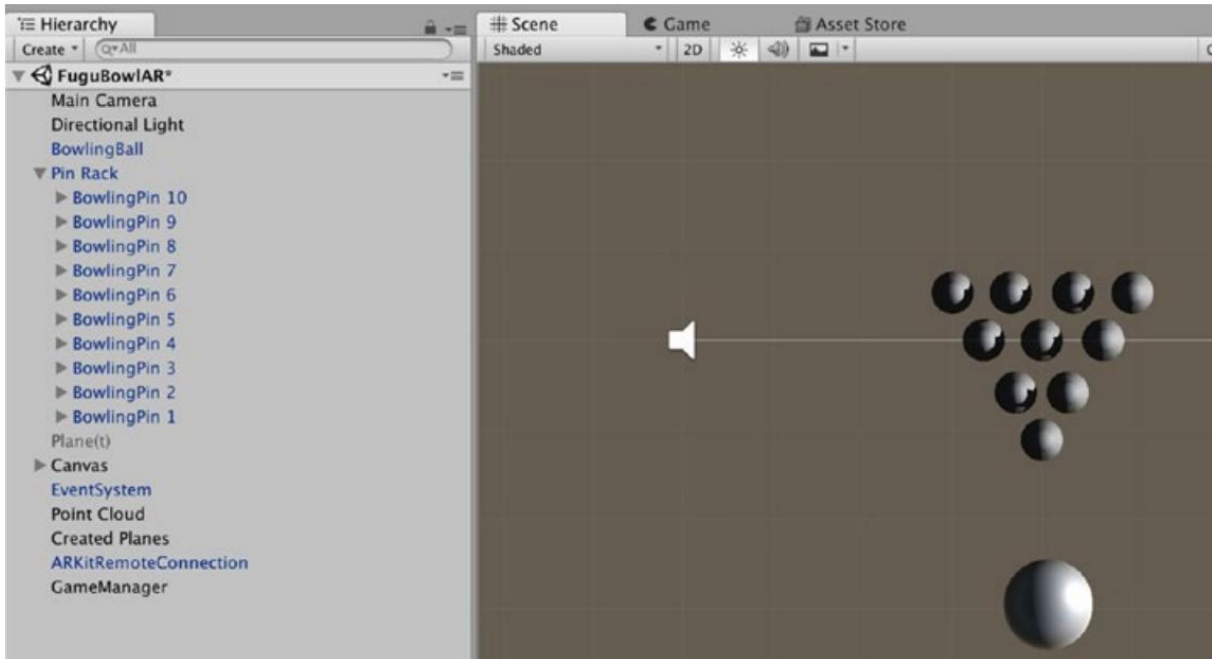
Meskipun ada dimensi yang benar untuk mengatur penempatan Pin Rack, kita akan menempatkan pin dalam segitiga terbalik tetapi akan menambahkan sedikit gaya artistik pada desain kita.

Dalam Hierarchy, pilih GameObject Bowling_Pin pertama dan ganti nama ini menjadi BowlingPin 5, dan pada Inspector pada pengaturan Prefab, pilih Apply. Meskipun tidak sepenuhnya diperlukan dalam contoh ini, kita akan membuat perubahan pada semua GameObject Bowling_Pin, jadi sebaiknya lakukan ini pada semuanya.

Dalam Hierarchy, pilih GameObject Bowling Pin kedua dan ganti namanya menjadi BowlingPin 4. Sekarang di Inspector, atur posisi transformasi X menjadi -1.12 dan kemudian pilih Apply untuk menerapkan semua perubahan pada Prefab ini. Kita akan mengikuti langkah yang sama untuk Pin Bowling yang tersisa. Namun, untuk menghemat waktu, saya telah merangkumnya dalam Tabel 8-1. Setelah selesai, Anda akan mendapatkan tampilan yang mirip dengan yang saya dapatkan pada Gambar 8-5.

Tabel 8.1. Pengaturan Pin Bowling

BowlingPin Number	X	Y	Z
BowlingPin 1	-2.4	0	0
BowlingPin 2	-0.5	0	-1.24
BowlingPin 3	0.5	0	-1.24
BowlingPin 4	-1.12	0	0
BowlingPin 5	0	0	0
BowlingPin 6	1.12	0	0
BowlingPin 7	1.72	0	1.12
BowlingPin 8	0.6	0	1.12
BowlingPin 9	0.6	0	1.12
BowlingPin 10	-2.4	0	1.12

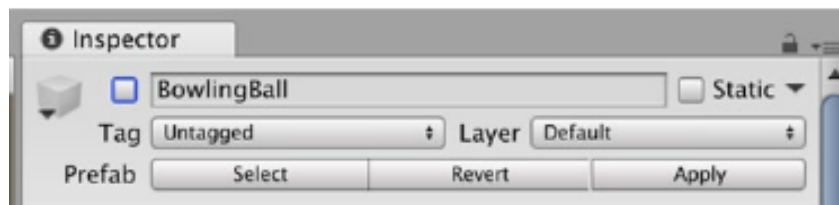


Gambar 8.5. Tampilan Pin Rack terakhir

Untuk sentuhan akhir, ganti nama GameObject BowlingPin Induk menjadi Pin Rack (yang secara teknis merupakan istilah yang tepat untuk segitiga terbalik Bowling Pin).

8.3 Membuat Instansi GameObject saat Runtime

Karena kita akan membuat instans BowlingBall dan BowlingPin Prefab saat runtime, kita dapat menonaktifkan aset yang relevan dalam Hierarchy. Pilih GameObject BowlingBall dan di Inspector, hapus centang pada kotak di samping nama Prefab (Gambar 8.6).

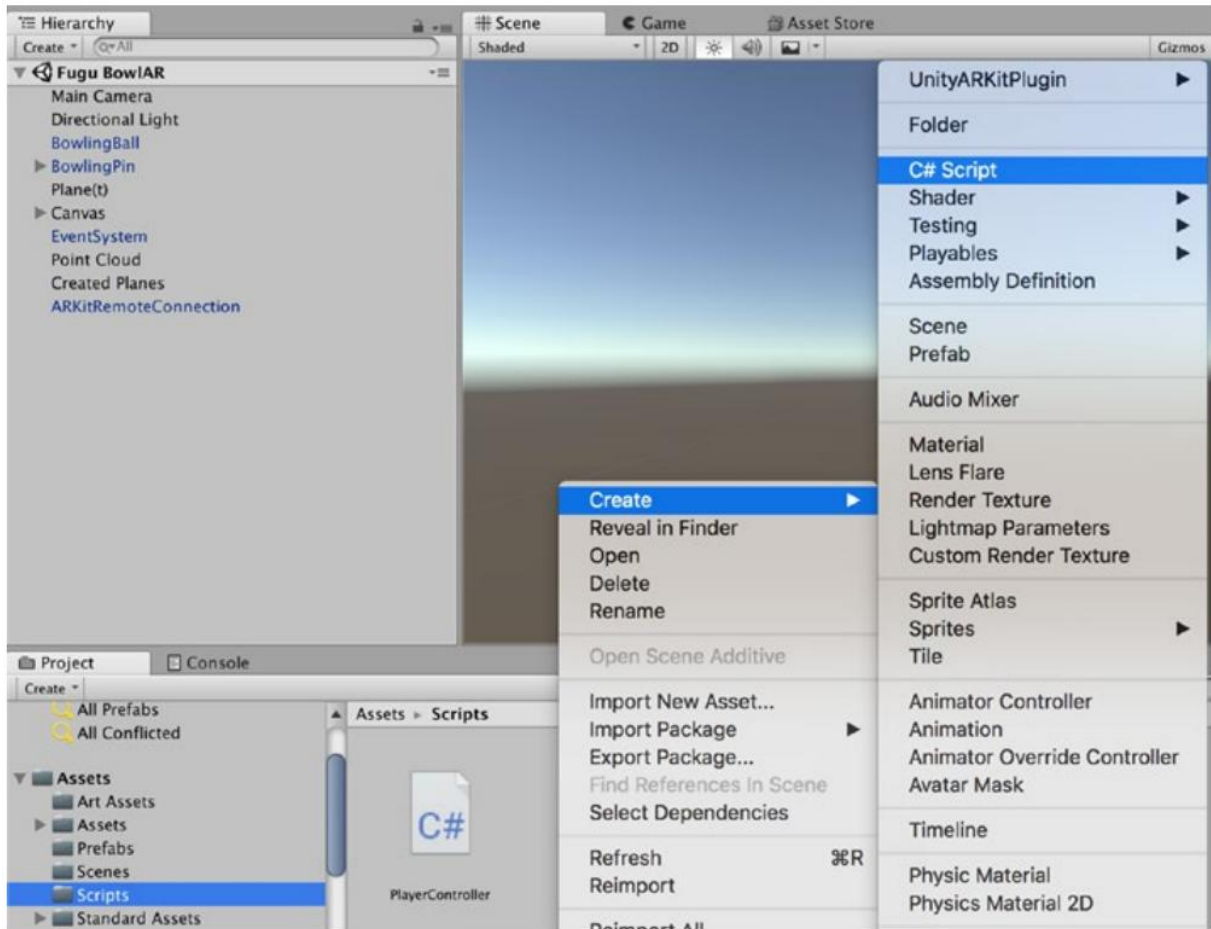


Gambar 8.6. Membatalkan pilihan BowlingBall dari Scene

Anda akan melihat bahwa BowlingBall Prefab tidak lagi terlihat dalam tampilan Scene. Ulangi langkah yang sama untuk Pin Rack.

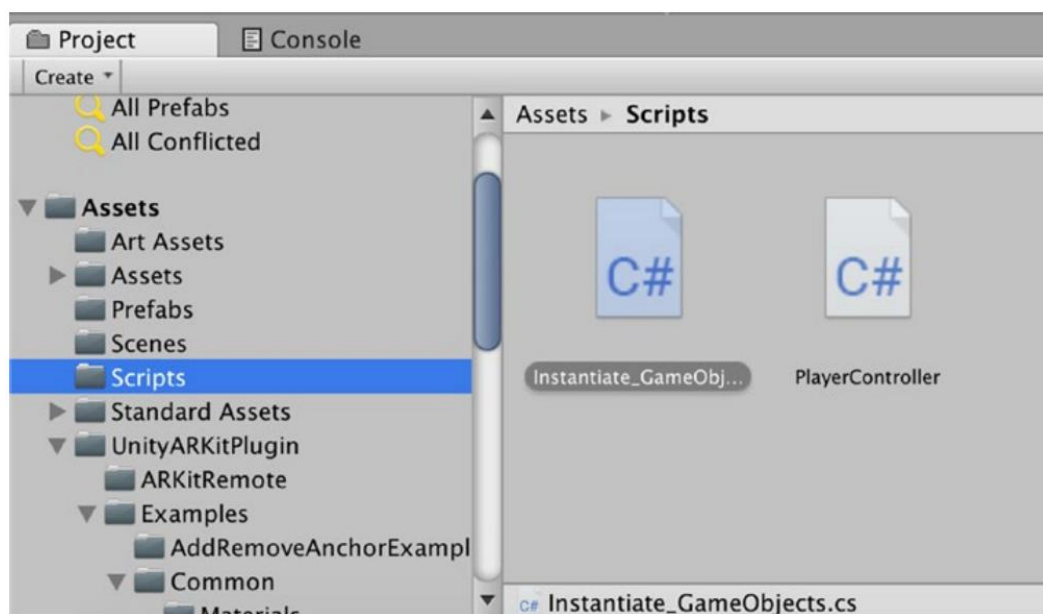
Membuat Script Instantiate_GameObjects

Jika Anda lupa cara membuat skrip, saya akan mengulangi langkah-langkah ini di sini. Jika Anda sudah tahu, silakan lewati beberapa paragraf berikutnya. Dalam tampilan Project, pilih folder Scripts dan klik kanan lalu pilih Create ► C# Script (Gambar 8-7).



Gambar 8.7. Membuat Skrip C#

Dengan Skrip baru yang dipilih, ganti nama skrip ini menjadi `Instantiate_GameObjects` (Gambar 8.8).



Gambar 8.8. File Skrip `Instantiate_GameObjects` dalam folder Skrip

Sekarang kita siap memasukkan skrip kita untuk membuat instance Prefab kita dalam game saat runtime. Klik dua kali file skrip `Instantiate_GameObjects` baru yang baru saja kita buat di IDE Anda, dan masukkan kode dalam Daftar 8-1.

Daftar 8-1. Skrip `Instantiate_GameObjects`

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Instantiate_GameObjects : MonoBehaviour
{
    public GameObject bowlingBallPrefab; //Used to store the
    bowling ball prefab and create an object based on that.

    public GameObject bowlingPinPrefab; //Used to store the
    bowling pin prefab and create an object based on that.

    public Transform terrain; //Here just to easily double check
    if we are finding the correct plane object.

    public bool once = false;

    // Update is called once per frame
    void LateUpdate()
    {
        //Look for a plane with the tag "Terrain" that will be
        generated based off another script and a prefab
        attached to it.
        terrain = GameObject.FindWithTag("Terrain").transform;
        if (once && (terrain != null))
        {
            once = false;
            InstantiateGameObjects(terrain.position);
        }
    }

    //Create a Bowling Ball and Pin Rack slightly above the
    plane/terrain/bowling lane.
    //Called from the ARGeneratePlane script, so that immediately
    after the plane is created, the game objects are too with
    minimal resource usage.
```

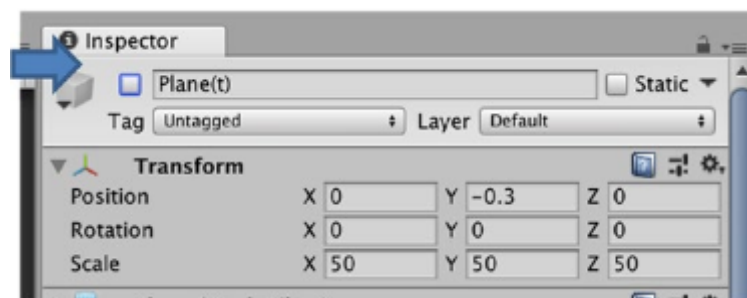
```

public void InstantiateGameObjects(Vector3 plane)
{
    once = false;
    plane.y += 0.5f;
    //Make sure that the objects appear above the plane.
    //Make sure the the objects appear towards the middle of
    the plane.
    if(plane.x >= 0)
    {
        plane.x += 2;
    }
    else
    {
        plane.x += -2;
    }
    Instantiate(bowlingBallPrefab, plane, Quaternion.
    identity); //Create the Pin Rack
}
}

```

Menonaktifkan GameObject dari Scene

Jika Anda tidak yakin cara menonaktifkan GameObject, pilih GameObject Plane(t) di Hierarchy, dan di Inspector, batalkan pilihannya dengan mengklik kotak centang tepat di sebelah nama GameObject (Gambar 8-9).



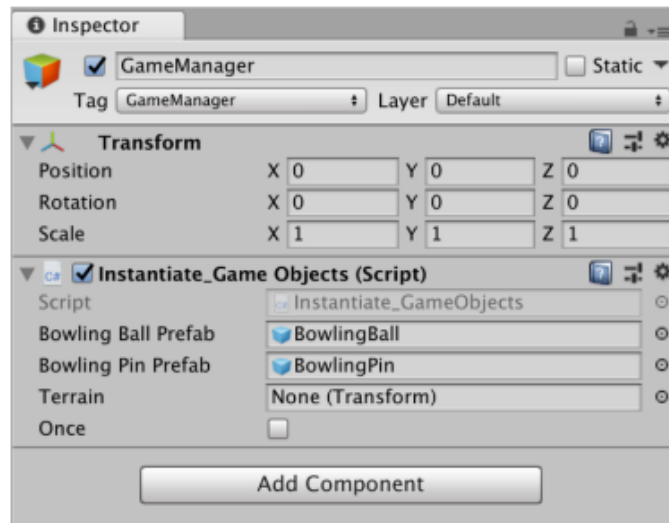
Gambar 8-9. GameObject yang dinonaktifkan

8.4 MEMBUAT GAME MANAGER

Sekarang kita akan membuat Game Manager. Game Manager adalah tempat yang bagus untuk menyimpan aset seperti Sistem Penilaian, manajer Audio, dan tempat yang bagus untuk menampung skrip Instantiate_GameObjects yang kita buat.

Di Hierarchy, buat GameObject Kosong dan beri nama GameManager. Dari Folder Skrip di

tampilan Proyek, seret skrip Instantiate_GameObjects ke GameObject ini (Gambar 8.10).



Gambar 8.10. GameManager dengan skrip Instantiate_GameObjects

Pada Gambar 8.10, saya telah menambahkan BowlingBall dan BowlingPin Prefab ke GameManager.

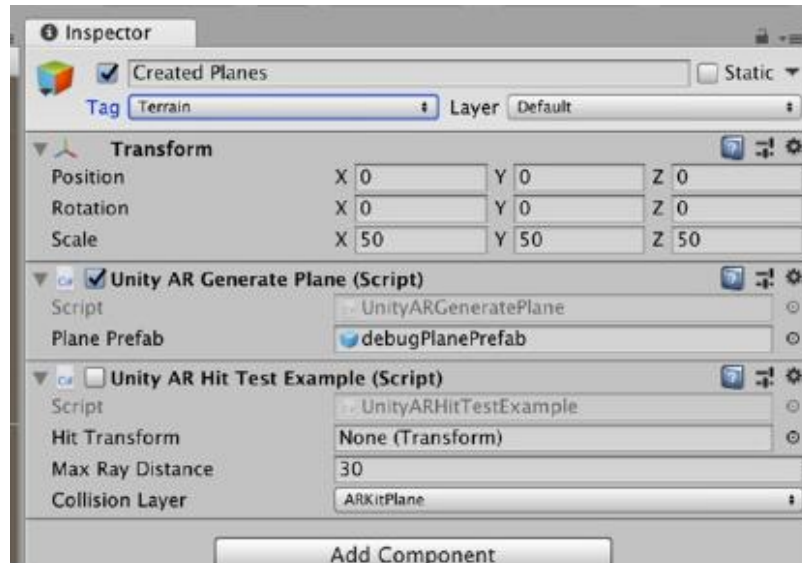
FindWithTag

Pembaca yang cermat akan melihat bahwa dalam Skrip Instantiate_GameObjects, kami merujuk ke fungsi FindWithTag. Kami perlu memperbarui GameObject Created Planes untuk menambahkan tag ini. Di Hierarchy, pilih GameObject Created Planes dan di Inspector, pilih menu drop-down dari Tag Properties dan Pilih Tag Terrain (Gambar 8.11).



Gambar 8.11. Memilih Tag Medan untuk GameObject Bidang yang Dibuat

Karena kita tidak lagi menggunakan Komponen Pengujian Hit, kita dapat menonaktifkannya (Gambar 8.12).



Gambar 8-12. Pengaturan GameObject Bidang yang Dibuat

Untuk game kita, kita tidak benar-benar memerlukan Point Cloud, jadi saya juga akan menonaktifkannya dari adegan. Sekarang saatnya untuk menyimpan game kita.

8.5 PENGUJIAN

Dalam dunia pengembangan game (dan pengembangan aplikasi), kita tidak dapat menghindari pengujian dan pengujian ulang. Namun, kali ini, saya akan membangun dan mencoba menjalankan game hebat saya di perangkat saya. Hal pertama yang perlu kita lakukan adalah menonaktifkan GameObject ARKitRemote di adegan. Saya tidak akan menghapusnya dulu, karena saya rasa kita mungkin harus melakukan lebih banyak pengujian dan pengembangan.

Di Hierarki, pilih Game Object ARKitRemote dan batalkan pilihannya.

Sekarang saatnya untuk menyimpan game kita. Saya telah memutuskan untuk menggunakan kontrol versi untuk nama file adegan saya. Dengan cara ini, jika terjadi kesalahan, saya dapat kembali ke adegan yang berhasil. Oleh karena itu, dari menu, saya memilih

File ► Save As dan saya memilih untuk memberi nama file Scene saya FuguBowlAR Test, dan saya menyimpannya di folder Scene.

Sekarang dari menu Utama, pilih Build settings. Pastikan Anda telah memilih scene saat ini dan di Player Settings, ubah nomor versinya.

Sekarang pilih Build and Run.

Wah, berhasil. Sekarang kita dapat menghapus semua GameObjects dan komponen yang tidak

kita gunakan dan mengujinya lagi. Jika tidak berhasil, periksa pengaturan Anda dan coba lagi. Berhasil bagi saya, dan saya akui, ini masih jauh dari game yang dapat dikirim sepenuhnya, tetapi ini adalah tempat yang bagus untuk memulai.

Ringkasan

Dalam bab ini, kami mengonversi dua GameObject menjadi prefab. Kemudian kami membuat dan memperbarui BowlingBall Prefab. Kami kemudian menambahkan skrip untuk membuat Instansiasi GameManager Script yang akan menambahkan prefab ini ke dalam game saat runtime.

DAFTAR PUSTAKA

- Apple Inc. (2017). *ARKit: Augmented reality for iOS*. Apple Developer. <https://developer.apple.com/arkit/>
- Baird, L. (2020). *Hands-On AR with Unity: Creating Augmented Reality Games and Applications for iOS*. Wiley.
- Bates, A. (2020). *Developing Augmented Reality Apps with ARKit and Unity*. Packt Publishing.
- Bates, B. (2019). *Unity 2018 Augmented Reality and Virtual Reality Game Development*. Packt Publishing.
- Beitler, S. (2020). *Unity for Mobile Game Development: Game Programming with Unity and C#*. Cengage Learning.
- Bell, J. (2020). *The Complete Unity ARKit Game Development Guide: Create Interactive AR Games with Unity and ARKit for iOS*. Wiley.
- Bos, A., & Smith, R. (2019). *Mastering Unity 2018: Augmented Reality Game Development*. Packt Publishing.
- Card, J. (2018). *Beginning iOS 12 Programming with Swift: Developing Apps for iPhone and iPad*. Apress.
- Carney, T. (2020). *Augmented Reality: A Practical Guide for Developers*. Packt Publishing.
- Clark, L. (2018). *Learning ARKit for iOS Development: Augmented Reality for iPhone and iPad*. O'Reilly Media.
- Clarkson, B. (2020). *Introduction to ARKit: Augmented Reality with Unity and iOS Development*. Packt Publishing.
- Collingwood, J. (2020). *Mastering Unity 2019 Game Development: Advanced Game Development Techniques*. Packt Publishing.
- Duffy, A. (2020). *Learning Unity for iOS Development: A Beginner's Guide to Game Development with Unity and C#*. O'Reilly Media.
- Eng, R., & Leong, A. (2019). *Augmented Reality for Mobile Devices: A Beginner's Guide to ARKit and Unity*. O'Reilly Media.
- Fields, J. (2019). *Unity 2018 Game Development in AR & VR: Designing Immersive Experiences for Mobile and VR*. Apress.
- Foster, H. (2020). *Developing Immersive Games with Unity: AR and VR Game Development for iOS and Android*. Packt Publishing.

- Gibson, P. (2019). *Developing Augmented Reality with Unity 3D and ARKit*. Packt Publishing.
- Gibson, S. (2020). *Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR*. Apress.
- Hall, C., & Brown, E. (2020). *The Unity ARKit Handbook: Building Augmented Reality Apps for iOS with Unity and Swift*. Packt Publishing.
- Hans, P., & Tran, H. (2020). *Building AR Games for iOS with Unity and ARKit*. Pearson Education.
- Harrison, P., & Hopkins, A. (2018). *Mobile Game Development with Unity: A Complete Guide to Designing, Developing, and Deploying Mobile Games*. Apress.
- Houghton, T. (2020). *Hands-On Augmented Reality with Unity 2019: A Beginner's Guide to Developing AR Applications*. Packt Publishing.
- Johnston, M. (2020). *Unity AR Game Development: An Introduction to Augmented Reality with Unity and ARKit for iOS*. Wiley.
- Jonsson, H. (2020). *Mastering ARKit: Augmented Reality for iOS Development with Swift and Unity*. Packt Publishing.
- Kessler, P., & MacNaughton, T. (2020). *Exploring Augmented Reality with Unity 3D: Augmented Reality Game Development with ARKit and Unity*. Wiley.
- King, B. (2019). *Unity 2020 AR and VR Game Development: Mastering Mobile AR/VR Game Development with Unity for iOS*. O'Reilly Media.
- King, R. (2019). *Augmented Reality Game Development with ARKit and Unity: Develop Stunning Augmented Reality Games for iOS*. Apress.
- Koch, J. (2020). *Building Augmented Reality with Unity and ARKit: Getting Started with AR Development on iOS*. Apress.
- Leach, R. (2020). *Developing AR Games with Unity 2019 for iOS and Android*. Apress.
- Li, X. (2020). *Building AR Games on iOS with Unity: Mastering Augmented Reality Game Development*. Apress.
- Margetson, R. (2020). *Developing Augmented Reality Applications with Unity 3D*. Packt Publishing.
- McDonald, P. (2020). *Unity 2020 AR and VR Game Development: Creating Augmented Reality Games with Unity*. O'Reilly Media.
- McGraw, J., & Piers, F. (2020). *Unity for iOS: Mastering ARKit and Building Interactive AR Games*. Wiley.
- Newton, B. (2019). *Building iOS AR Games with Unity: Developing Augmented Reality Games for iPhone and iPad*. Packt Publishing.

- O'Donnell, M. (2019). *Unity AR and VR Game Development: Create Interactive Augmented and Virtual Reality Experiences*. Packt Publishing.
- Pearson, T. (2020). *ARKit Essentials: Augmented Reality for iOS Development*. Packt Publishing.
- Pettigrew, L. (2019). *Creating Augmented Reality with Unity and ARKit for iOS Development*. Cengage Learning.
- Pugh, A. (2020). *Hands-On ARCore: Building Augmented Reality Apps with Android and Unity*. Packt Publishing.
- Ross, G. (2020). *The ARKit Handbook: Augmented Reality Development with Unity for iOS*. O'Reilly Media.
- Scully, M., & Clark, J. (2017). *Building Augmented Reality with Unity and ARKit: A Developer's Guide to Augmented Reality in iOS*. John Wiley & Sons.
- Shaw, J. (2020). *Hands-On Augmented Reality with Unity: A Guide to Building Augmented Reality Applications*. Packt Publishing.
- Smith, C. (2019). *Unity 3D Game Development for iOS and Android: Developing 3D Games with Unity and ARKit*. Packt Publishing.
- Smith, D. (2020). *Introduction to Augmented Reality and ARKit: Develop AR Experiences on iOS with Swift and Unity*. Springer.
- Smith, R. (2020). *Advanced iOS AR Development: Exploring Augmented Reality with ARKit and Unity*. O'Reilly Media.
- Smithson, J. (2019). *Unity 3D for Mobile AR Development: A Beginner's Guide to Augmented Reality on iOS*. Packt Publishing.
- Stokes, R., & Taylor, A. (2020). *Unity 2019 Mobile Game Development: Build Compelling Games for iOS and Android*. Apress.
- Tabrizian, B. (2020). *Unity 2019 for iOS Development: Build Games and Apps with Unity and ARKit*. Packt Publishing.
- Taylor, D. (2019). *Building Augmented Reality Applications with Unity and ARKit: Developing Immersive AR Experiences for iOS*. Packt Publishing.
- Taylor, R. (2020). *ARKit 2.0 for iOS Development: Create Augmented Reality Apps with Unity and ARKit*. O'Reilly Media.
- Thomas, A. (2020). *Augmented Reality for Beginners: Unity 3D and ARKit for iOS Development*. Wiley.
- Thomas, D. (2020). *ARKit for iOS Development: A Beginner's Guide to Augmented Reality and Game Creation with Unity*. Apress.

- Thompson, G. (2020). *Unity 2019 Augmented Reality Game Development: Building AR Games for iOS*. O'Reilly Media.
- Turner, J. (2020). *Building Augmented Reality Apps for iOS with ARKit and Unity*. Packt Publishing.
- Tyler, P. (2020). *Mastering Augmented Reality for iOS with Unity and ARKit*. Apress.
- Unity Technologies. (2018). *Unity Manual - ARKit Plugin*.
<https://docs.unity3d.com/Manual/ARKit.html>
- Wallace, J., & Griffith, J. (2019). *Unity 2018 AR and VR Game Development: Building Augmented and Virtual Reality Games for iOS and Android*. Packt Publishing.
- Watkins, D. (2020). *Unity ARKit: Develop AR Apps and Games for iOS with Unity and ARKit*. Wiley.
- Wendel, R. (2018). *Unity 2018 AR and VR Development: Building Mobile AR Games with Unity*. Packt Publishing.
- Westerman, R. (2020). *Creating AR Applications for iOS with Unity*. Wiley.
- Williams, M. (2018). *Practical Unity for AR/VR Development*. O'Reilly Media.
- Williams, S. (2019). *Creating Augmented Reality Games with ARKit and Unity: A Practical Guide to Game Development on iOS*. Springer.
- Wilson, P. (2020). *Unity 2020 AR and VR Game Development: Mastering Mobile AR/VR Game Development with Unity*. Packt Publishing.
- Wright, F. (2020). *Building Augmented Reality Games for iOS with ARKit and Unity*. Apress.
- Wright, R. (2019). *Learning Unity 2018 for Augmented and Virtual Reality Game Development*. Apress.
- Zhao, L. (2019). *Creating AR Applications with ARKit and Unity: A Practical Guide for iOS Developers*. Wiley.



Pengembangan GAME AR (Augmented Reality)

Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Bio Data Penulis



Penulis lahir di Semarang pada tanggal 1 Maret 1983. Penulis menempuh pendidikan Sarjana Teknik Elektro di Universitas Kristen Satya Wacana (UKSW), lulus tahun 2004, kemudian tahun 2005 melanjutkan studi pada Magister Desain di Fakultas Seni Rupa dan Desain, Institut Teknologi Bandung (ITB), dan kemudian melanjutkan studi pada program studi

Teknologi Multimedia di Swinburne University of Technology Australia. Penulis sejak tahun 2010, menjadi dosen pada program studi Desain Grafis Universitas Sains dan Teknologi Komputer (Universitas STEKOM), memiliki Jabatan Akademik Lektor Kepala 700. Penulis juga seorang wirausaha di bidang toko online yang berhasil di kota Semarang dan juga aktif sebagai freelancer dalam bidang fotografi, web design dan multimedia.



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :

YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-623-8642-64-9 (PDF)



9

786238

642649